

Iterative ensemble smoother as an approximate solution to a regularized minimum-average-cost problem: theory and applications

Xiaodong Luo · Andreas S. Stordal · Rolf J. Lorentzen · Geir Nævdal

Abstract The focus of this work is on an alternative implementation of the iterative ensemble smoother (iES). We show that iteration formulae similar to those used in [3,6] can be derived by adopting a regularized Levenberg-Marquardt (RLM) algorithm [14] to approximately solve a minimum-average-cost (MAC) problem. This not only leads to an alternative theoretical tool in understanding and analyzing the behaviour of the aforementioned iES, but also provides insights and guidelines for further developments of the smoothing algorithms. For illustration, we compare the performance of an implementation of the RLM-MAC algorithm to that of the approximate iES used in [3] in three numerical examples: an initial condition estimation problem in a strongly nonlinear system, a facies estimation problem in a 2D reservoir and the history matching problem in the Brugge field case. In these three specific cases, the RLM-MAC algorithm exhibits comparable or even better performance, especially in the strongly nonlinear system.

Introduction

For data assimilation problems there are different ways in utilizing the available observations. While certain data assimilation algorithms, for instance, the ensemble Kalman filter (EnKF, see, for example, [2,8]), assimilate the observations sequentially in time, other data assimilation algorithms may instead

Xiaodong Luo · Andreas S. Stordal · Rolf J. Lorentzen · Geir Nævdal

International Research Institute of Stavanger (IRIS), 5008 Bergen, Norway

E-mail: xiaodong.luo@iris.no

collect the observations at different time instants and assimilate them simultaneously. Examples in this aspect include the ensemble smoother (ES, see, for example, [9]) and its iterative variants.

The EnKF has been widely used for reservoir data assimilation (history matching) problems since its introduction to the community of petroleum engineering [26]. The applications of the ES to reservoir data assimilation problems are also investigated recently (see, for example, [33]). Compared to the EnKF, the ES has certain technical advantages, including, for instance, avoiding the restarts associated with each update step in the EnKF for certain reservoir simulators (e.g., ECLIPSE[®] [1]) and also having fewer variables to update. The formal benefit (avoiding restarts) may result in a significant reduction of simulation time in certain circumstances [33], while the latter (having fewer variables) can reduce the amount of computer memory in use.

To further improve the performance of the ES, some iterative ensemble smoothers (iES) are suggested in the literature, in which the iterations are carried out in the forms of certain iterative optimization algorithms, e.g., the Gaussian-Newton [4, 10, 16] or the Levenberg-Marquardt method [3, 6, 13, 18, 21, 22]¹, or in the context of adaptive Gaussian mixture (AGM, see [34]). In [13, 22], the iteration formulae are adopted following the regularized Levenberg-Marquardt (RLM) algorithm in the deterministic inverse problem theory (see, for example, [7, 15]). Essentially these formulae aim to find a single solution of the inverse problem, and the gradient involved in the iteration is obtained either through the adjoint model [13], or through a stochastic approximation method [22]. While in [6], the iteration formula is derived based on the idea that, the final result of the iES should be equal to the estimate of the EnKF, at least for linear systems with Gaussian model and observation errors. Consequently, this algorithm is called *ensemble smoother with multiple data assimilation* (ES-MDA for short). On the other hand, in [3] an iteration formula is obtained based on the standard Levenberg-Marquardt (LM) algorithm. By discarding a model term in the standard LM algorithm, an approximate iteration formula is derived, which is similar to that in [6]. For distinction, we call it approximate Levenberg-Marquardt ensemble randomized maximum likelihood (aLM-EnRML) algorithm.

¹ These optimization algorithms are also applied to iterative EnKFs, e.g., in some of the aforementioned works.

In this work we show that an iteration formula similar to those used in the ES-MDA and RLM-MAC can be derived by applying the RLM to find an ensemble of solutions via solving a minimum-average-cost problem [21]. The gradient involved in the iteration is obtained in a way similar to the computation of the Kalman gain matrix in the EnKF. This derivation not only leads to an alternative theoretical tool in understanding and analyzing the behaviour of the aforementioned iES, but also provides insights and guidelines for further developments of the iES algorithm². As an example, we derive an alternative implementation of the iterative ES based on the RLM algorithm. Three numerical examples are then used to illustrate the performance of this new algorithm and compare it to the aLM-EnRML.

Methodologies

ES-MDA and aLM-EnRML

Let \mathbf{m} denote an m -dimensional reservoir model that contains the petrophysical properties to be estimated, \mathbf{g} the reservoir simulator, and \mathbf{d}^o a p -dimensional vector that contains all available observations in a certain time interval. Here \mathbf{d}^o is assumed to contain certain measurement errors, with zero mean and covariance \mathbf{C}_d . In the context of iterative ensemble smoothing, suppose that there is an ensemble $\mathbf{M}^i \equiv \{\mathbf{m}_j^i\}_{j=1}^{N_e}$ of N_e reservoir models available at the i th iteration step, then an iES updates \mathbf{M}^i to its counterpart $\mathbf{M}^{i+1} \equiv \{\mathbf{m}_j^{i+1}\}_{j=1}^{N_e}$ at the next iteration step via a certain iteration formula, which is the focus of our discussion below. For convenience of discussion later, let us define the following square root matrix \mathbf{S}_m^i with respect to the model \mathbf{m} (*model square root* for short):

$$\mathbf{S}_m^i = \frac{1}{\sqrt{N_e - 1}} [\mathbf{m}_1^i - \bar{\mathbf{m}}^i, \dots, \mathbf{m}_{N_e}^i - \bar{\mathbf{m}}^i], \quad \bar{\mathbf{m}}^i = \frac{1}{N_e} \sum_{j=1}^{N_e} \mathbf{m}_j^i, \quad (1)$$

in the sense that the product $\mathbf{S}_m^i (\mathbf{S}_m^i)^T$ is equal to the sample covariance of \mathbf{M}^i .

The idea in the ES-MDA is to make the final iES estimate equal the estimate of the EnKF, at least for linear systems with Gaussian model and observation errors (some thoughts are also provided in Appendix A in situations with nonlinearity and/or non-Gaussianity). To this end, an iteration formula is derived

² For instance, instead of using the RLM algorithm, one could apply other types of regularized inversion methods, e.g., the regularized Gaussian-Newton or conjugate gradient algorithm [7, 15], to solve the minimum-average-cost problem. This will thus lead to more types of iterative ES algorithms.

in terms of [6]

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \mathbf{S}_m^i (\mathbf{S}_d^i)^T \left(\mathbf{S}_d^i (\mathbf{S}_d^i)^T + \gamma^i \mathbf{C}_d \right)^{-1} (\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_j^i)), \quad i = 1, 2, \dots, I; \quad (2)$$

$$\mathbf{S}_d^i = \frac{1}{\sqrt{N_e - 1}} \left[\mathbf{g}(\mathbf{m}_1^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)}, \dots, \mathbf{g}(\mathbf{m}_{N_e}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right]; \quad (3)$$

$$\overline{\mathbf{g}(\mathbf{m}_j^i)} = \frac{1}{N_e} \sum_{j=1}^{N_e} \mathbf{g}(\mathbf{m}_j^i). \quad (4)$$

The total number of iteration steps I is chosen before the iteration starts; \mathbf{d}_j^o ($j = 1, \dots, N_e$) are the perturbations of \mathbf{d}^o , similar to those in the EnKF; and γ^i ($i = 1, \dots, I$) satisfy

$$\gamma^i \geq 1 \quad \text{and} \quad \sum_{i=1}^I \frac{1}{\gamma^i} = 1. \quad (5)$$

For convenience, we also call \mathbf{S}_d^i *data square root* hereafter, in the sense that $\mathbf{S}_d^i (\mathbf{S}_d^i)^T$ is equal to the sample covariance matrix of the simulated data $\{\mathbf{g}(\mathbf{m}_j^i)\}_{j=1}^{N_e}$.

In the Levenberg-Marquardt ensemble randomized maximum likelihood (LM-EnRML) method [3], history matching is recast as a weighted least-squares problem, in the sense that each model, say \mathbf{m}_j^i (the j th ensemble model at the i th iteration step), is updated to the one \mathbf{m}_j^{i+1} at the next iteration step by solving the following minimization problem:

$$\underset{\mathbf{m}}{\operatorname{argmin}} (\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}))^T \mathbf{C}_d^{-1} (\mathbf{d}_j^o - \mathbf{g}(\mathbf{m})) + (\mathbf{m} - \mathbf{m}_j^0)^T (\mathbf{C}_m^0)^{-1} (\mathbf{m} - \mathbf{m}_j^0), \quad (6)$$

with $\mathbf{C}_m^0 = \mathbf{S}_m^0 (\mathbf{S}_m^0)^T$, and \mathbf{S}_m^0 given by Eq. (1). Applying the Levenberg-Marquardt algorithm [24] to (6), one has the following iteration formula to update the model [3]

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \Delta \mathbf{m}_j^i, \quad (7)$$

$$\begin{aligned} \Delta \mathbf{m}_j^i &= \left[(\mathbf{C}_m^0)^{-1} + (\mathbf{G}_j^i)^T \mathbf{C}_d^{-1} \mathbf{G}_j^i \right]^{-1} \left[(\mathbf{G}_j^i)^T \mathbf{C}_d^{-1} (\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_j^i)) - (\mathbf{C}_m^0)^{-1} (\mathbf{m}_j^i - \mathbf{m}_j^0) \right] \\ &\approx \left[(1 + \lambda^i) (\mathbf{C}_m^i)^{-1} + (\mathbf{G}_j^i)^T \mathbf{C}_d^{-1} \mathbf{G}_j^i \right]^{-1} \left[(\mathbf{G}_j^i)^T \mathbf{C}_d^{-1} (\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_j^i)) - (\mathbf{C}_m^0)^{-1} (\mathbf{m}_j^i - \mathbf{m}_j^0) \right], \end{aligned}$$

where $\mathbf{C}_m^i = \mathbf{S}_m^i (\mathbf{S}_m^i)^T$, \mathbf{G}_j^i is the Jacobian of \mathbf{g} evaluated at \mathbf{m}_j^i , and λ^i a positive scalar. In the course of deriving the third line of Eq. (7), the first $(\mathbf{C}_m^0)^{-1}$ is approximated by $(1 + \lambda^i) (\mathbf{C}_m^i)^{-1}$, in accordance to the Levenberg-Marquardt algorithm [3]. In large-scale problems, it can be expensive to evaluate the term $(\mathbf{C}_m^0)^{-1} (\mathbf{m}_j^i - \mathbf{m}_j^0)$, because the model size m is usually much larger than the observation size p (i.e., $m \gg p$). Hence discarding the term $(\mathbf{C}_m^0)^{-1} (\mathbf{m}_j^i - \mathbf{m}_j^0)$, using the Sherman-Morrison-Woodbury formula [32] and some algebra, one has

$$\mathbf{m}_j^{i+1} \approx \mathbf{m}_j^i + \mathbf{C}_m^i (\mathbf{G}_j^i)^T \left[\mathbf{G}_j^i \mathbf{C}_m^i (\mathbf{G}_j^i)^T + (1 + \lambda^i) \mathbf{C}_d \right]^{-1} (\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_j^i)). \quad (8)$$

As to be shown later (see Eq. (15)), with a suitable choice for λ^i , the iteration formula in Eq. (8) is consistent with the one derived from the RLM algorithm. Under suitable technical conditions (see, for example, [14]), \mathbf{m}_j^i will converge to a solution of the equation $\mathbf{g}(\mathbf{m}) = \mathbf{d}_j^o$ as $i \rightarrow +\infty$.

Following the EnKF, one may introduce a further approximation to the gain matrix in Eq. (8), such that

$$\mathbf{C}_m^i (\mathbf{G}_j^i)^T \left[\mathbf{G}_j^i \mathbf{C}_m^i (\mathbf{G}_j^i)^T + (1 + \lambda^i) \mathbf{C}_d \right]^{-1} \approx \mathbf{S}_m^i (\mathbf{S}_d^i)^T \left(\mathbf{S}_d^i (\mathbf{S}_d^i)^T + (1 + \lambda^i) \mathbf{C}_d \right)^{-1}. \quad (9)$$

Therefore, with the above approximations, one has the following final iteration formula

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \mathbf{S}_m^i (\mathbf{S}_d^i)^T \left(\mathbf{S}_d^i (\mathbf{S}_d^i)^T + (1 + \lambda^i) \mathbf{C}_d \right)^{-1} (\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_j^i)). \quad (10)$$

Comparing Eqs. (10) and (2), it is clear that the iteration formulae become identical when $(1 + \lambda^i)$ in Eq. (10) is replaced by γ^i in Eq. (2).

For distinction, we call the algorithm in Eq. (10) *approximate Levenberg-Marquardt ensemble randomized maximum likelihood* (aLM-EnRML) method. In the aLM-EnRML, starting from an initial value λ^0 , the subsequent values of λ^i will decrease if the average data mismatch with respect to the models is reduced, otherwise the values of λ^i will increase instead. The iteration process will stop if the maximum iteration number is reached, or if the relative change of the average data mismatch in two consecutive iteration steps is lower than a given threshold.

The regularized Levenberg-Marquardt (RLM) algorithm and an extension

In the sequel we first introduce the regularized Levenberg-Marquardt (RLM, as used in [13, 22]) algorithm in the context of deterministic inverse problems theory, and then consider an extension of the RLM algorithm as an ensemble data assimilation method. In the course of deduction, the similarities between the ES-MDA/aLM-EnRML and the extended method in the current study will become clear, while some implications in the extended method will also be discussed. Note that in a recent work [12], the author also considers using the RLM algorithm in the context of ensemble smoothing. Compared to [12], the current study has different focuses, e.g., in terms of the way in approximating the gradient of the cost function of the minimization problem, and the parameter rule used in the RLM algorithm (see (16) later)³. Readers are referred to [12] for more detail.

³ To summarize, [12, 13, 22] and the current work are all related to the RLM algorithm. [12, 13] use the parameter rule based on, for example, the work [11], while [22] and the current work use the parameter rule based on [14]. Although different in the concrete forms, both parameter rules are proven to lead to local convergence of the RLM algorithm [11, 14]. In addition, [13, 22] both aim to find a single solution, but employ different methods in approximating the gradient of the

In the conventional deterministic inverse problem theory (see, for example, [7, 15]), one aims to find a single set of parameters (e.g., a reservoir model \mathbf{m}) whose simulated output (e.g., $\mathbf{g}(\mathbf{m})$) matches the observation \mathbf{d}^o . Intuitively, one can search for such a model by solving the following weighted least-squares problem

$$\underset{\mathbf{m}}{\operatorname{argmin}} (\mathbf{d}^o - \mathbf{g}(\mathbf{m}))^T \mathbf{C}_d^{-1} (\mathbf{d}^o - \mathbf{g}(\mathbf{m})), \quad (11)$$

which aims to minimize the *data mismatch* $(\mathbf{d}^o - \mathbf{g}(\mathbf{m}))^T \mathbf{C}_d^{-1} (\mathbf{d}^o - \mathbf{g}(\mathbf{m}))$, where \mathbf{C}_d^{-1} is the weight matrix associated with the data mismatch term. For convenience of discussion later, let us define $\|\mathbf{d}\|_2 \equiv \sqrt{\mathbf{d}^T \mathbf{d}}$ and $\|\mathbf{d}\|_{\mathbf{C}_d} \equiv \sqrt{\mathbf{d}^T \mathbf{C}_d^{-1} \mathbf{d}}$ for a vector \mathbf{d} , then it is clear that $(\mathbf{d}^o - \mathbf{g}(\mathbf{m}))^T \mathbf{C}_d^{-1} (\mathbf{d}^o - \mathbf{g}(\mathbf{m})) = \|\mathbf{d}^o - \mathbf{g}(\mathbf{m})\|_{\mathbf{C}_d}^2 = \|\mathbf{C}_d^{-1/2} (\mathbf{d}^o - \mathbf{g}(\mathbf{m}))\|_2^2$.

When the dimension of the observation space is lower than that of the model space (as is often the case in practice), (11) is an under-determined and ill-posed problem, which has some undesirable features, e.g., non-uniqueness of the solution and potentially large sensitivity of the solution to the observation, in the sense that even a very small change in the observation might lead to a large change in the solution [7]. To mitigate the above problems, in deterministic inverse problems theory it is customary to introduce a certain regularization term to (11). Here we consider Tikhonov regularization, in which a regularization term, in terms of $(\mathbf{m} - \mathbf{m}^b)^T \mathbf{C}_m^{-1} (\mathbf{m} - \mathbf{m}^b)$, is introduced to (11), such that the weighted least-squares problem becomes

$$\underset{\mathbf{m}}{\operatorname{argmin}} (\mathbf{d}^o - \mathbf{g}(\mathbf{m}))^T \mathbf{C}_d^{-1} (\mathbf{d}^o - \mathbf{g}(\mathbf{m})) + \gamma (\mathbf{m} - \mathbf{m}^b)^T \mathbf{C}_m^{-1} (\mathbf{m} - \mathbf{m}^b), \quad (12)$$

where γ is a positive scalar, \mathbf{m}^b denotes a background model, and \mathbf{C}_m^{-1} is the associated weight matrix. The choice of γ affects the relative weights assigned to the data mismatch term and the regularization term, and thus influences the solution of (12). For instance, if γ is relatively large (e.g., tending to ∞ in the extreme case), then the obtained solution will approach \mathbf{m}^b . On the other hand, if γ is relatively small (e.g., tending to 0), then (11) and (12) tend to coincide with each other, and the solution of (12) can be considered as an approximation to one of the solutions of (11). Apart from γ , the matrices \mathbf{C}_m and \mathbf{C}_d would also affect the relative weights between the regularization and data mismatch terms.

If, in addition, \mathbf{C}_m and \mathbf{C}_d are chosen to be the error covariance matrices of the background and the cost function, as discussed previously. In contrast, [12] and the current study target for multiple solutions in the context of iES, and differentiate each other in terms of the ways in constructing the data square root matrices (e.g., [12] uses the same way as in the EnKF). In Appendix A, we also provide an alternative point of view to interpret the iES algorithm derived in the current study.

observations, respectively, then the correlations of the variables in the model and observation spaces, respectively, are also incorporated in the minimization problem.

In general, one may wish to choose a γ such that the data mismatch of the resulting solution is comparable to the noise level of the observations [7, 15]. The true noise level is often unknown, therefore in practice it may be replaced by, for instance, a threshold value proportional to p (i.e., the size of \mathbf{d}^o). Readers are referred to, for example, [13, 23, 30] for the rationales behind this choice.

For linear systems, there are straightforward ways for one to construct a solution that has the desired data mismatch (see, for example, [23]). For nonlinear systems, however, one often has to rely on a certain iterative algorithm to find the solution. One of such methods, called the regularized Levenberg-Marquardt algorithm following [14] (or the regularizing Levenberg-Marquardt algorithm following [11]), constructs a sequence of model $\{\mathbf{m}^i\}$ ($i = 1, 2, \dots$) by solving a linearized weighted least-squares problem

$$\begin{aligned} \underset{\mathbf{m}^{i+1}}{\operatorname{argmin}} & (\mathbf{d}^o - \mathbf{g}(\mathbf{m}^i) - \mathbf{G}_i(\mathbf{m}^{i+1} - \mathbf{m}^i))^T \mathbf{C}_d^{-1} (\mathbf{d}^o - \mathbf{g}(\mathbf{m}^i) - \mathbf{G}_i(\mathbf{m}^{i+1} - \mathbf{m}^i)) \\ & + \gamma^i (\mathbf{m}^{i+1} - \mathbf{m}^{i+1,b})^T \mathbf{C}_m^{-1} (\mathbf{m}^{i+1} - \mathbf{m}^{i+1,b}), \end{aligned} \quad (13)$$

at each iteration step. In (13), \mathbf{G}_i is the Jacobian of \mathbf{g} evaluated at \mathbf{m}^i , such that $\mathbf{g}(\mathbf{m}^i) + \mathbf{G}_i(\mathbf{m}^{i+1} - \mathbf{m}^i)$ represents a first order Taylor approximation to $\mathbf{g}(\mathbf{m}^{i+1})$; γ^i is a positive scalar as in (12), but now changes over the iteration; $\mathbf{m}^{i+1,b}$ is the background model that may also be adaptive with the iteration. A convenient choice can be $\mathbf{m}^{i+1,b} = \mathbf{m}^i$, meaning that we want the new model \mathbf{m}^{i+1} not to be too far away from the previous model, such that the approximation $\mathbf{g}(\mathbf{m}^{i+1}) \approx \mathbf{g}(\mathbf{m}^i) + \mathbf{G}_i(\mathbf{m}^{i+1} - \mathbf{m}^i)$ can be roughly valid [22]. In addition, the choice $\mathbf{m}^{i+1,b} = \mathbf{m}^i$ also simplifies the iteration formula, as will be shown below.

The solution of the weighted least-squares problem (13) is given by

$$\begin{aligned} \mathbf{m}^{i+1} &= \mathbf{m}^i + \Delta \mathbf{m}^i \\ \Delta \mathbf{m}^i &= \left[\gamma^i \mathbf{C}_m^{-1} + (\mathbf{G}^i)^T \mathbf{C}_d^{-1} \mathbf{G}^i \right]^{-1} \left[(\mathbf{G}^i)^T \mathbf{C}_d^{-1} (\mathbf{d}^o - \mathbf{g}(\mathbf{m}^i)) - \gamma^i \mathbf{C}_m^{-1} (\mathbf{m}^i - \mathbf{m}^{i+1,b}) \right], \end{aligned} \quad (14)$$

which is similar to Eq. (7) (e.g., by replacing $1 + \lambda^i$ therein with γ^i). By letting $\mathbf{m}^{i+1,b} = \mathbf{m}^i$ and with some algebra, one has the simplified iteration formula

$$\mathbf{m}^{i+1} = \mathbf{m}^i + \mathbf{C}_m (\mathbf{G}^i)^T \left[\mathbf{G}^i \mathbf{C}_m (\mathbf{G}^i)^T + \gamma^i \mathbf{C}_d \right]^{-1} (\mathbf{d}^o - \mathbf{g}(\mathbf{m}^i)). \quad (15)$$

which is thus similar to Eq. (10). One may also let the weight matrix \mathbf{C}_m be adaptive with the iteration, which will be considered later in the context of iES. Intuitively, provided that the step size $\Delta \mathbf{m}^i$ is small

enough (which can be controlled by γ^i) such that the first order Taylor expansion of $\mathbf{g}(\mathbf{m}^{i+1})$ around \mathbf{m}^i is a roughly valid approximation of $\mathbf{g}(\mathbf{m}^{i+1})$, one has

$$\begin{aligned}
\|\mathbf{g}(\mathbf{m}^{i+1}) - \mathbf{d}^o\|_{\mathbf{C}_d}^2 &\approx \|\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o + \mathbf{G}^i(\mathbf{m}^{i+1} - \mathbf{m}^i)\|_{\mathbf{C}_d}^2 \\
&= \|[\mathbf{I} - \mathbf{G}^i \mathbf{C}_m (\mathbf{G}^i)^T [\mathbf{G}^i \mathbf{C}_m (\mathbf{G}^i)^T + \gamma^i \mathbf{C}_d]^{-1}] (\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o)\|_{\mathbf{C}_d}^2 \\
&= \|\gamma^i \mathbf{C}_d [\mathbf{G}^i \mathbf{C}_m (\mathbf{G}^i)^T + \gamma^i \mathbf{C}_d]^{-1} (\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o)\|_{\mathbf{C}_d}^2 \\
&= \left(\mathbf{C}_d^{-1/2} (\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o) \right)^T \left[\gamma^i \left[\left(\mathbf{C}_d^{-1/2} \mathbf{G}^i \right) \mathbf{C}_m (\mathbf{C}_d^{-1/2} \mathbf{G}^i)^T + \gamma^i \mathbf{I}_p \right]^{-1} \right]^2 \left(\mathbf{C}_d^{-1/2} (\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o) \right) \\
&= \|\gamma^i \left[\left(\mathbf{C}_d^{-1/2} \mathbf{G}^i \right) \mathbf{C}_m (\mathbf{C}_d^{-1/2} \mathbf{G}^i)^T + \gamma^i \mathbf{I}_p \right]^{-1} \left(\mathbf{C}_d^{-1/2} (\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o) \right)\|_2^2 \\
&\leq \|\gamma^i \left[\left(\mathbf{C}_d^{-1/2} \mathbf{G}^i \right) \mathbf{C}_m (\mathbf{C}_d^{-1/2} \mathbf{G}^i)^T + \gamma^i \mathbf{I}_p \right]^{-1}\|_2^2 \|\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o\|_{\mathbf{C}_d}^2 \\
&\leq \|\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o\|_{\mathbf{C}_d}^2,
\end{aligned}$$

since $\|\gamma^i \left[\left(\mathbf{C}_d^{-1/2} \mathbf{G}^i \right) \mathbf{C}_m (\mathbf{C}_d^{-1/2} \mathbf{G}^i)^T + \gamma^i \mathbf{I}_p \right]^{-1}\|_2^2 \leq 1$, where \mathbf{I}_p denotes the p -dimensional identity matrix, and $\|\mathbf{A}\|_2$ represents the spectral norm of the matrix \mathbf{A} . This implies that the data mismatch $\|\mathbf{g}(\mathbf{m}^{i+1}) - \mathbf{d}^o\|_{\mathbf{C}_d}^2$ at the $(i+1)$ -th iteration tends to be no larger than the one $\|\mathbf{g}(\mathbf{m}^i) - \mathbf{d}^o\|_{\mathbf{C}_d}^2$ at the previous iteration.

In terms of the choice of γ^i , the following parameter rule from [14]

$$\begin{aligned}
\gamma^0 &> 0, \\
\gamma^{i+1} &= \rho \gamma^i, \text{ with } 1/r < \rho < 1 \text{ for some scalar } r > 1, \\
\lim_{i \rightarrow +\infty} \gamma^i &= 0,
\end{aligned} \tag{16}$$

can be used, in which the scalar sequence $\{\gamma^i\}$ gradually reduces to zero as i tends to $+\infty$, while the presence of the lower bound $1/r$ for the coefficient ρ aims to prevent any abrupt drop-down of γ^i to zero. When Eq. (15) is used in conjunction with (16), it can be analytically shown that the data mismatch of the sequence of models $\{\mathbf{m}^i\}$ converges to zero locally as $i \rightarrow +\infty$ and the observation noise level tends to zero, provided that the equation $\mathbf{g}(\mathbf{m}) = \mathbf{d}^o$ is solvable and some other conditions are satisfied [14]. Instead of using (16), one can adopt other parameter rules that also lead to local convergence of the data mismatch under similar technical conditions (see, for example, [11]).

When the observation noise is present, however, it may not be desirable to have a too small data mismatch in order to prevent over-fitting the observations. In general, one may wish to let the iteration stop when the data mismatch of the current iterated model is comparable to the noise level, which is often referred to as the *discrepancy principle* [7, 15]. Of course, in practice the true noise level is typically unknown. As a result, one may let the iteration process Eq. (15) stop when either of the following three

conditions are satisfied: (a) the data mismatch of the iterated model is lower than a pre-set threshold $\beta_u^2 p$ for the first time for a given positive scalar β_u , in light of the discussions in [13, 22, 23, 30]; or (b) the maximum iteration number is reached; or (c) the relative change of the data mismatch in two consecutive iteration steps is lower than a given threshold (0.01% in our implementation). Note that, with the stopping condition (a), the total number of (actual) iteration steps can be less than the maximum iteration number, for instance, when the RLM reaches the threshold $\beta_u^2 p$ earlier than up to the maximum iteration number. Conditions (b) and (c) are the same as those in the aLM-EnRML, and are mainly introduced to control the runtime of the iteration process. These settings will be applied to the aLM-EnRML and our extension scheme in the experiments later, in order to make the comparison of both algorithms be conducted under the same experimental settings as far as possible.

In the practical implementation of the RLM algorithm (and our extension scheme below), there are a few issues that need to be taken into account. Firstly, in the theoretical analysis of the RLM algorithm (see, for example, [7, 11, 14, 15]), it typically assumes that the initial model stays sufficiently close to a solution of the equation $\mathbf{g}(\mathbf{m}) = \mathbf{d}^o$, which is not necessarily true in reality. In addition, as can be seen in our previous discussion, the validity of the first order Taylor approximation $\mathbf{g}(\mathbf{m}^{i+1}) \approx \mathbf{g}(\mathbf{m}^i) + \mathbf{G}_i(\mathbf{m}^{i+1} - \mathbf{m}^i)$ is essential to the derivation of the RLM algorithm. In practice, it is often not a trivial issue to obtain the Jacobian \mathbf{G}_i . If the exact Jacobian \mathbf{G}_i is not available, then one must evaluate it based on a certain approximation scheme, which often has to struggle between the computational efficiency and accuracy. Putting these factors together, in practice it is likely that the iteration process, Eq. (15), may occasionally lead to higher data mismatch for the current iterated model, in comparison to that of the previous one, due to, for example, a too large step size that invalidates the first order Taylor approximation. In such cases, following [3], a simple remedy can then be to increase the parameter γ^i to some extent and re-execute the iteration in Eq. (15) (which will then have a smaller step size). This strategy is essentially similar to the back-tracking line search method in optimization theory (see, for example, [28]) and works well in our numerical studies.

In the remainder of this section we consider an extended RLM scheme in the context of ensemble smoothing. As in the aLM-EnRML, instead of examining the data mismatch of a single model, we are more interested in studying how the average data mismatch of the ensemble models changes over the

iteration step. To this end, the weighted least-squares problem in (12) is replaced with the following minimum-average-cost (MAC) problem, namely,

$$\begin{aligned} \underset{\{\mathbf{m}_j^{i+1}\}_{j=1}^{N_e}}{\operatorname{argmin}} \quad & \frac{1}{N_e} \sum_{j=1}^{N_e} \left[\left(\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_j^{i+1}) \right)^T \mathbf{C}_d^{-1} \left(\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_j^{i+1}) \right) \right. \\ & \left. + \gamma^i \left(\mathbf{m}_j^{i+1} - \mathbf{m}_j^i \right)^T \left(\tilde{\mathbf{C}}_m^i \right)^{-1} \left(\mathbf{m}_j^{i+1} - \mathbf{m}_j^i \right) \right], \end{aligned} \quad (17)$$

where $\tilde{\mathbf{C}}_m^i = \tilde{\mathbf{S}}_m^i \left(\tilde{\mathbf{S}}_m^i \right)^T$ ($\tilde{\mathbf{S}}_m^i$ will be specified later), and \mathbf{d}_j^o , \mathbf{m}_j^i and \mathbf{m}_j^{i+1} are the same as those defined previously. For distinction, we call the resulting iES *regularized Levenberg-Marquardt algorithm based on minimum-average-cost* (RLM-MAC for short). Appendix A also includes some thoughts that aim to interpret the RLM-MAC from the point of view of an expectation-maximization algorithm [5].

To solve (17), it is necessary to linearize the forward model \mathbf{g} as in the RLM algorithm. Note that in (17) there are N_e simulated observation terms $\mathbf{g}(\mathbf{m}_j^{i+1})$ ($j = 1, 2, \dots, N_e$). To reduce the computational cost in evaluating the Jacobian matrices, one may choose to linearize the forward model \mathbf{g} , for each ensemble member \mathbf{m}_j^{i+1} , around a “common” point, say \mathbf{m}_c^i , such that

$$\mathbf{g}(\mathbf{m}_j^{i+1}) \approx \mathbf{g}(\mathbf{m}_c^i) + \mathbf{G}_c^i \left(\mathbf{m}_j^{i+1} - \mathbf{m}_c^i \right), \quad (18)$$

where \mathbf{G}_c^i is the Jacobian of \mathbf{g} at \mathbf{m}_c^i . Inserting Eq. (18) into (17) and with some algebra (see Appendix B), one has the following iteration formula

$$\begin{aligned} \mathbf{m}_j^{i+1} &= \mathbf{m}_j^i + \Delta \mathbf{m}_j^i \\ \Delta \mathbf{m}_j^i &= \tilde{\mathbf{S}}_m^i \left(\mathbf{G}_c^i \tilde{\mathbf{S}}_m^i \right)^T \left[\left(\mathbf{G}_c^i \tilde{\mathbf{S}}_m^i \right) \left(\mathbf{G}_c^i \tilde{\mathbf{S}}_m^i \right)^T + \gamma^i \mathbf{C}_d \right]^{-1} \left(\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_c^i) - \mathbf{G}_c^i \left(\mathbf{m}_j^i - \mathbf{m}_c^i \right) \right), \end{aligned} \quad (19)$$

for $j = 1, \dots, N_e$. If one lets

$$\tilde{\mathbf{S}}_m^i = \frac{1}{\sqrt{N_e - 1}} \left[\mathbf{m}_1^i - \mathbf{m}_c^i, \dots, \mathbf{m}_{N_e}^i - \mathbf{m}_c^i \right], \quad (20)$$

and adopts the approximation $\mathbf{G}_c^i \left(\mathbf{m}_j^i - \mathbf{m}_c^i \right) \approx \mathbf{g}(\mathbf{m}_j^i) - \mathbf{g}(\mathbf{m}_c^i)$, then Eq. (19) reduces to

$$\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \tilde{\mathbf{S}}_m^i \left(\tilde{\mathbf{S}}_d^i \right)^T \left(\tilde{\mathbf{S}}_d^i \left(\tilde{\mathbf{S}}_d^i \right)^T + \gamma^i \mathbf{C}_d \right)^{-1} \left(\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_c^i) \right), \quad (21)$$

where

$$\tilde{\mathbf{S}}_d^i = \frac{1}{\sqrt{N_e - 1}} \left[\mathbf{g}(\mathbf{m}_1^i) - \mathbf{g}(\mathbf{m}_c^i), \dots, \mathbf{g}(\mathbf{m}_{N_e}^i) - \mathbf{g}(\mathbf{m}_c^i) \right]. \quad (22)$$

Compared to the iteration formulae of ES-MDA and aLM-EnRML (e.g., Eq. (10)), the iteration formula of RLM-MAC in general differs in the way of constructing the square root matrices $\tilde{\mathbf{S}}_m^i$ and $\tilde{\mathbf{S}}_d^i$. Possible choices for \mathbf{m}_c^i could be, for instance, the mean $\bar{\mathbf{m}}^i$ or a member (e.g., the one closest to $\bar{\mathbf{m}}^i$) of the ensemble $\{\mathbf{m}_j^i\}_{j=1}^{N_e}$. In the current work we let $\mathbf{m}_c^i = \bar{\mathbf{m}}^i$.

Under the choice $\mathbf{m}_c^i = \bar{\mathbf{m}}^i$, $\tilde{\mathbf{S}}_m^i$ coincides with \mathbf{S}_m^i , therefore the resulting RLM-MAC algorithm mainly differs from the aLM-EnRML in the data square root matrix. In general, $\tilde{\mathbf{S}}_d^i$ and \mathbf{S}_d^i may have different “centering points”: in Eq. (3), the simulated observations $\mathbf{g}(\mathbf{m}_j^i)$ in \mathbf{S}_d^i center around $\overline{\mathbf{g}(\mathbf{m}_j^i)}$, while in $\tilde{\mathbf{S}}_d^i$ they center around $\mathbf{g}(\bar{\mathbf{m}}^i)$ instead. In certain circumstances, e.g., when the forward model \mathbf{g} is linear or weakly nonlinear, $\overline{\mathbf{g}(\mathbf{m}_j^i)}$ and $\mathbf{g}(\bar{\mathbf{m}}^i)$ may be close to each other⁴. In some other cases, the distinction between $\overline{\mathbf{g}(\mathbf{m}_j^i)}$ and $\mathbf{g}(\bar{\mathbf{m}}^i)$ may be more substantial, and the behaviour of aLM-EnRML and RLM-MAC may thus become more diverged, as will be shown later. In terms of the performance comparison between the aLM-EnRML and RLM-MAC, however, we stress that, although one may see the relative superiority of one algorithm over the other in some specific cases, it is expected that the conclusion might not be valid in a broader sense, following the “no free lunch theorems for optimization” [37]. Intuitively, since both the aLM-EnRML and RLM-MAC are local optimization algorithms and they may have different search directions, these two algorithms might end up with different local optima in general, and the relative superiority between these two algorithms may thus vary from case to case.

Finally we note that, in the course of solving (17), we have made use of the first order Taylor approximation around a common point \mathbf{m}_c^i in order to reduce the computational cost. In this regard, a theoretically more sound – yet computationally more expensive – strategy would be to linearize the forward model \mathbf{g} , for each ensemble model, around a certain point in its neighbourhood. A trade-off strategy may also be employed, e.g., by clustering the ensemble models into a small number of groups, and then linearizing \mathbf{g} around a common point for each group. In addition, in the context of solving a MAC problem, it is also possible for one to explicitly incorporate into (17) additional terms that impose certain constraints onto the statistics of the ensemble members or the corresponding simulated observations, such that the iteration step may be in a form beyond the Kalman update formula as in Eq. (21). An investigation in this aspect will be carried out in the future.

⁴ In particular, when \mathbf{g} is linear, $\overline{\mathbf{g}(\mathbf{m}_j^i)} = \mathbf{g}(\bar{\mathbf{m}}^i)$, such that the RLM-MAC and aLM-EnRML become identical.

Numerical examples

After discussing the similarities and differences between the ES-MDA, aLM-EnRML and RLM-MAC in the previous section, we focus on demonstrating potentially different behaviour of the ensemble smoothers that is resulted from using different “centering points” $\overline{\mathbf{g}(\mathbf{m}_j^i)}$ or $\mathbf{g}(\bar{\mathbf{m}}^i)$ to construct the data square root matrix.

As noted previously, in the ES-MDA, the total number I of iteration steps is decided before the iteration process starts, and the parameters γ^i are dependent on I , in light of the constraint (5). The aLM-EnRML and RLM-MAC do not have the above features in general: Under the stopping conditions (a-c) in the previous section, the total numbers of the iteration steps of the aLM-EnRML and RLM-MAC depend on the threshold $\beta_u^2 p$ and the convergence speed of the algorithms in specific problems, subject to the constraint of maximum iteration number. In addition, the parameters γ^i are chosen according to a different rule, which is independent of I or the maximum iteration number in general. Therefore, for our purpose here, we confine ourselves to the comparison between the aLM-EnRML and the RLM-MAC. In the implementations, both algorithms apply the truncated singular value decomposition (TSVD) to the data square root matrices (normalized by $\mathbf{C}_d^{-1/2}$) in their iteration formulae⁵, and retain the leading eigenvalues which sum up to at least 99% of the “total energies” (see [3,36] for the details). In the experiments below we first examine the performance of both algorithms in a strongly nonlinear system, and then apply both algorithms to a reservoir facies estimation problem and the Brugge field case.

An initial state estimation problem in a strongly nonlinear system

Here we adopt the strongly nonlinear 40-dimensional Lorenz 96 (L96) model [19] to demonstrate the potentially different behaviour between the aLM-EnRML and RLM-MAC. The governing equations of

⁵ In doing so, the iteration formulae, e.g., that in Eq. (21), are re-expressed as $\mathbf{m}_j^{i+1} = \mathbf{m}_j^i + \tilde{\mathbf{S}}_m^i (\mathbf{C}_d^{-1/2} \tilde{\mathbf{S}}_d^i)^T \left((\mathbf{C}_d^{-1/2} \tilde{\mathbf{S}}_d^i) (\mathbf{C}_d^{-1/2} \tilde{\mathbf{S}}_d^i)^T + \gamma^i \mathbf{I}_p \right)^{-1} \mathbf{C}_d^{-1/2} \left(\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_j^i) \right)$, where \mathbf{I}_p is the p -dimensional identity matrix. The TSVD is applied to the product $(\mathbf{C}_d^{-1/2} \tilde{\mathbf{S}}_d^i) (\mathbf{C}_d^{-1/2} \tilde{\mathbf{S}}_d^i)^T$, which can have certain numerical benefits, e.g., mitigating the issue of different orders of amplitudes in the measurement data.

the L96 model are given by

$$\frac{dx_k}{dt} = (x_{k+1} - x_{k-2})x_{k-1} - x_k + F, \quad k = 1, \dots, 40. \quad (23)$$

In Eq. (23) the variables x_k are defined in a cyclic way such that $x_{-1} = x_{39}$, $x_0 = x_{40}$ and $x_{41} = x_1$. The L96 model is designed to mimic baroclinic turbulence in the midlatitude atmosphere. Because of its strongly nonlinear nature, this model is often employed to test the performance of data assimilation algorithms in the weather data assimilation community.

In this work we let $F = 8$, with which the L96 model is chaotic. The L96 model is discretized through the fourth-order Runge-Kutta method with a constant integration step of 0.05 (corresponding to 6 hours in real time, see [19]). To initialize the model, we draw an initial state vector at random from the multivariate normal distribution $N(\mathbf{0}, \mathbf{I}_{40})$, where \mathbf{I}_{40} is the 40-dimensional identity matrix. To avoid the transit effect, the model states within the first 500 integration steps are discarded, and data assimilation starts after this “spin-up” period. For reference, hereafter we re-label the model state at time step 500 as the initial state, which has an impact on subsequent model states and their observations. In the experiment the initial state vector is assumed unknown, and will be estimated based on the observations at subsequent time steps.

In the experiment the assimilation time window is 10 days, and the observations are available every 24 hours. At each observation time instant (e.g., 24h, 48h etc.), the synthetic observations are generated by first applying the nonlinear function $f(x) = x^3/5$ to the odd number state variables x_1, x_3, \dots, x_{39} (overall 20 variables), and then adding a sample from the normal distribution $N(0, 1)$ to each function value $f(x_k)$ ($k = 1, 3, \dots, 39$). The total number of observation elements in the assimilation time window is thus $10 \times 20 = 200$.

The initial background ensemble is drawn from a multivariate normal distribution whose mean and covariance are obtained from the “climatological” statistics of the L96 model in a long free-run with 100,000 integration steps (see, for example, [22] for the details). The ensemble size of the aLM-EnRML is 100. Note that in the RLM-MAC one also needs to compute $\mathbf{g}(\bar{\mathbf{m}}^i)$, the simulated observation of the ensemble mean, apart from those of the ensemble members. Because of this, we let the ensemble size of

the RLM-MAC be 99, such that the aLM-EnRML and RLM-MAC have the same number of simulated observations at each iteration step. The maximum number of iterations is 100, and the coefficient $\beta_u = 2$.

In light of the requirement that the first order Taylor approximation be roughly valid, it is suggested to start with a relatively small step size at the beginning [22]. As a result, the initial value γ^0 should be relatively large (but not too large in order to reduce the total number of iteration steps). In our implementation, we let $\gamma^i = \alpha^i \sqrt{\text{trace}(\mathbf{S}_d^i (\mathbf{S}_d^i)^T) / N_e}$, where $\text{trace}(\bullet)$ calculates the trace of a matrix, $\alpha^0 = 1$ and $\alpha^{i+1} = 0.9 \alpha^i$ if the average data mismatch of the ensemble is reduced, otherwise the i th iteration step is repeated, with $\alpha^i = 2 \alpha^i$, for maximum 5 times, following [3]. In the subsequent two other case studies, the same way is applied to tune α^i adaptively. We stress that this should not be considered as the optimal rule. Instead, it is expected that one may improve it for specific applications in general.

As the dimension of the L96 model is relatively low, we can afford to repeat the experiment 100 times. In each repetition, the initial background ensemble and the associated observations are drawn at random. In the same repetition run, the aLM-EnRML and RLM-MAC have the same initial background ensemble, the associated observations and the parameter rule in choosing γ^i .

Figures 1 - 3 report the data mismatch and RMSEs of the aLM-EnRML and RLM-MAC, obtained from 100 repetitions of the experiment⁶. In terms of the data mismatch, from Figures 1 and 2, one can see that for the aLM-EnRML, its range is roughly between 10^3 and 10^6 . Overall, 2% of the data mismatch lies in the interval $[10^3, 10^4]$, 51% in the interval $[10^4, 10^5]$, and 47% in the interval $[10^5, 10^6]$. For the RLM-MAC, the range of the data mismatch is roughly between 10^2 and 10^6 . there are 11% of the data mismatch contained in the interval $[10^2, 10^3]$, 13% in the interval $[10^3, 10^4]$, 51% in the interval $[10^4, 10^5]$, and the remaining 25% in the interval $[10^5, 10^6]$. This suggests that in some cases the RLM-MAC will have data mismatch lower than the minimum one of the aLM-EnRML. Similar results can also be observed in terms of the RMSEs (Figures 1 and 3). As can be seen in Figure 1, for the aLM-EnRML, its RMSEs seem to follow a uni-modal distribution, with its (single) peak around 3. On the other hand, for the RLM-MAC, the distribution of its RMSEs tends to be more flat, with one of its peaks being closer to zero. By counting the numbers of the RMSEs in the sub-intervals (e.g., $[0, 1]$, $[1, 2]$ etc) of the range $[0, 6]$, the pie

⁶ Here the data mismatch is calculated according to (11), and is averaged over all ensemble members. The RMSE of a m -dimensional estimate $\hat{\mathbf{v}}$ with respect to its truth \mathbf{v}^{tr} is give by $\|\hat{\mathbf{v}} - \mathbf{v}^{tr}\|_2 / \sqrt{m}$, where $\|\bullet\|_2$ denotes the ℓ_2 -norm.

charts in Figure 3 indicate that, the percentages of the RMSEs of the aLM-EnRML in the sub-intervals (from low to high values) are 2%, 18%, 34%, 31%, 12% and 3%, respectively, while the percentages with respect to the RLM-MAC are 22%, 18%, 28%, 16%, 14% and 2%. Overall, Figures 1 and 3 also suggest that in some cases the RLM-MAC will have RMSEs lower than the minimum one of the aLM-EnRML.

Figure 4 shows the box plots of the data mismatch of the aLM-EnRML (left) and RLM-MAC (right) at different iteration steps in one repetition run. The ensemble average data mismatch in both iES decreases with the iteration. In this particular case, however, the data mismatch of the RLM-MAC tends to converge faster than that of the aLM-EnRML. In addition, its final data mismatch (up to the maximum iteration step in the figure) is substantially lower than that of the aLM-EnRML. We also note that, the top outlier in each box plot of the aLM-EnRML does not appear to follow the tendency of changes of other data mismatch values in the same box plot, which is a phenomenon not observed in the RLM-MAC. A possible explanation of this difference might be as follows: In the RLM-MAC, in order for the cost function in the MAC problem to decrease with respect to the change of each ensemble member (see the deduction in Appendix B), it is natural to use the deviations

$$\mathbf{g}(\mathbf{m}_j^i) - \mathbf{g}(\bar{\mathbf{m}}^i) \quad (24)$$

to construct the data square root matrix on top of the first order Taylor approximation. In the aLM-EnRML, however, (24) is replaced by

$$\mathbf{g}(\mathbf{m}_j^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)}. \quad (25)$$

The difference between (24) and (25) would then depend on the degree of nonlinearity in \mathbf{g} . In this particular case, because of the strong nonlinearity in the L96 model, the difference between (25) and (24) may be relatively large (see Figure 6 later for a numerical investigation), such that the iteration formula based on the choice (25) does not form gradient descent directions for all the ensemble members in the aLM-EnRML.

In accordance to the results in Figure 4, top panels of Figure 5 show the box plots of RMSEs of the ensemble members with respect to the initial conditions at different iteration steps in the aLM-

EnRML (left) and RLM-MAC (right), and bottom panels indicate the associated ensemble spreads⁷. For both the aLM-EnRML and RLM-MAC, their RMSEs drop at the first few iteration steps, bounce back subsequently, and then drop again until they converge (although a slight oscillation can also be observed around iteration step 85 in the aLM-EnRML). In this particular case, the final RMSEs of the RLM-MAC tend to be lower than those of the aLM-EnRML. The ensemble spread of the aLM-EnRML monotonically decreases with the iteration, while that of the RLM-MAC reaches its minimum around iteration step 10, and then slightly increases afterwards. The final ensemble spreads of both algorithms are close to each other, and they are both underestimated in comparison with the corresponding RMSEs. One possible reason of this underestimation phenomenon may be because in both smoothers the ensemble members tend to converge to regions surrounding certain local minima, therefore the ensemble spreads are relatively small, while the RMSEs are determined by the distances between the local minima and the truth, and thus are not necessarily as small as the corresponding ensemble spreads. Apart from this, we also envision that the effects of finite ensemble size (e.g., sampling errors and correlations among ensemble members [27]) may also contribute to the underestimation phenomenon.

Figure 6 depicts the normalized differences $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right)$ at different iteration steps, based on the corresponding ensemble members of the RLM-MAC. Please note that in this case, the box plots indicate the spatial distribution of the elements of the vector $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right)$, whose size is equal to the number of observations (rather than the ensemble size). In addition, in the box plots, $\mathbf{g}(\bar{\mathbf{m}}^i)$ and $\overline{\mathbf{g}(\mathbf{m}_j^i)}$ are computed from the same ensemble of the RLM-MAC at each iteration step, rather than one (e.g., $\mathbf{g}(\bar{\mathbf{m}}^i)$) from the ensemble of the RLM-MAC and another (e.g., $\overline{\mathbf{g}(\mathbf{m}_j^i)}$) from the corresponding ensemble of the aLM-EnRML.

From Figure 6 one can see that the spreads of the normalized differences are relatively large at the first few iteration steps. This may be because the initial ensemble spreads are relatively large (see the lower right panel of Figure 5), and the chaos nature of the L96 model tends to further amplify them. However, as the number of iterations increases, the ensemble members of the RLM-MAC tend to converge

⁷ Suppose that $\{\mathbf{m}_j\}_{j=1}^{N_e}$ is an ensemble of models with the ensemble size being N_e , and that $m_{j,s}$ is the s th element of the vector \mathbf{m}_j . Let σ_s^2 be the sample variance of the s th elements of all ensemble members, then the ensemble spread is defined as $\sqrt{\sum_{s=1}^m \sigma_s^2 / m}$, where m is the size of the ensemble members \mathbf{m}_j .

toward a certain local minimum, and the ensemble spread decreases rapidly. As a result, $\mathbf{g}(\bar{\mathbf{m}}^i)$ approach $\overline{\mathbf{g}(\mathbf{m}_j^i)}$, such that the normalized differences decrease and tend to center around zero.

A reservoir facies estimation problem

Here we consider a simple 2D facies model (see Figure 7) previously studied in [17]. The reservoir consists of 45×45 gridblocks, with only oil and water inside. The porosity of the reservoir is 0.1 in each gridblock, but the permeability varies spatially: It is 10000 md in the channels, and 500 md in the background. In the field there are 8 water injectors (I1 - I8) and 8 producers (P1 - P8), and their locations are marked as white dots in Figure 7. The injectors are constrained by injection rates ($15.9 \text{ m}^3/\text{day}$ in each), while the producers are constrained by bottom hole pressures (BHP, 399 bar in each).

The measurements used for history matching are the oil and water production rates in the producers and BHP in the injectors. To generate the synthetic measurements, we use the reference facies model in Figure 7 as the input to ECLIPSE[®] [1] and run the simulation for 1900 days. The measurements are then taken as the outputs of ECLIPSE[®] every 190 days, plus certain Gaussian noise (whose variances are $0.16 \text{ m}^3/\text{day}$ for production rates, and 0.07 bar for BHP). To enhance the connectivities of the estimated channels, eight statistical measures used in [17] are also assimilated into the model. These statistics measurements include the sample mean values of: (1) area of bodies (a body is defined as neighboring connected gridblocks with the same facies type, see [17]); (2) volume of bodies; (3) area-to-volume ratio of bodies; (4) maximal body extension in x-direction; (5) maximal body extension in y-direction; (6) 75th percentile for the extensions in the x-direction; (7) number of bodies; and (8) fraction of the total volume of the facies with respect to the total volume of the field, where all the mean values are evaluated with respect to the initial background ensemble [17]. Accordingly, in the course of assimilation, the measurement error variances associated with these statistics measurements are taken as their sample variances that are also evaluated with respect to the initial background ensemble. We have also run both the aLM-EnRML and RLM-MAC when the statistics measurements are not included in assimilation. In this case, the final data mismatch values are close to what we show in Figure 10 later, but the corresponding estimated channels tend to be less connected (results not shown), consistent with the observation in [17].

We follow [17] and use the signed distance function in the level set method to characterize the facies. When using an iES for history matching, there is no need to augment the model and the corresponding state variables as in the EnKF. Therefore for the problem considered here, the initial ensemble only contains the signed distance function values at each gridblock, which are converted from the facies models generated by SNESIM [35] through the MATLAB level set toolbox (version 1.1, [25]). As in the previous example, the ensemble size in the aLM-EnRML is 100, while that in the RLM-MAC is 99. In addition, the aforementioned parameter rule $\gamma^i = \alpha^i \sqrt{\text{trace}(\mathbf{S}_d^i (\mathbf{S}_d^i)^T)} / N_e$ is also adopted in this example. The maximum number of iterations is 20, and the coefficient $\beta_u = 2$.

A few initial models are shown in the first column of Figure 8 (a, d, g), and the corresponding models obtained at the final iteration steps of the aLM-EnRML and RLM-MAC are plotted in the second column (b, e, h) and the third one (c, f, i), respectively. As can be seen in these images, despite the clear difference among the initial models, all the updated models of the aLM-EnRML and RLM-MAC bear certain similarities to each other, and they also, to some extent, resemble the reference model in Figure 7.

Figure 9 reports the average scores of the initial ensemble (panel (a)) and the final ensembles of the aLM-EnRML (panel (b)) and RLM-MAC (panel (c)) in matching the facies of the reference model. The score is calculated as follows: at each gridblock, if a model matches the facies, and it has a score of 1, otherwise it has a score of 0. The final score is then averaged over all the models in the ensemble. Therefore a higher score means that more models have the correct estimated facies. From Figure 9 it is seen that, compared with the scores in the initial ensemble, it seems that the aLM-EnRML and RLM-MAC tend to improve the facies estimation in the channels, but also have deteriorated matching in certain areas (e.g., that around the injector I2). In addition, in this particular case, it seems that the RLM-MAC tends to perform better than the aLM-EnRML in terms of matching the facies.

Figure 10 indicates the data mismatch of the aLM-EnRML (left) and RLM-MAC (right) as functions of the number of iteration steps. The aLM-EnRML stops at the 10th iteration step due to the stopping condition (c) mentioned early (relative change of data mismatch less than 0.01%), while the RLM-MAC stops at the maximum iteration step. Despite this difference, in this case the aLM-EnRML and RLM-

MAC appear to have comparable performance, in terms of the convergence speed and the final data mismatch value. For the aLM-EnRML, its final data mismatch is 2197.71 ± 234.63 , where the number before \pm is the mean, and that after \pm is the standard deviations (STD). On the other hand, for the RLM-MAC, its final data mismatch is 1864.27 ± 98.9975 , slightly lower than that of the aLM-EnRML.

Following [17], Figures 11 and 12 show the performance of history matching oil and water production rates in producers P5 and P7, for the initial ensemble and the ensembles of the aLM-EnRML and RLM-MAC at their final iteration steps, respectively. In both figures, compared to the results with respect to the initial ensemble, those obtained from the final ensembles of the aLM-EnRML and RLM-MAC appear to have reduced ensemble spread and improved data mismatch. The aLM-EnRML performs better than the RLM-MAC in history matching producer P5, but the situation is reverted in producer P7. In the initial ensemble the oil production rates in P5 and the water production rates in P7 tend to be lower than the reference values (which may be related to the bias in the facies distribution of the initial ensemble, as highlighted in Figure 9(a)), and this bias trend remains in the final ensembles obtained by the aLM-EnRML and RLM-MAC. Similarly, in the initial ensemble the oil production rates in P7 and the water production rates in P5 tend to be higher than the reference values. This bias trend is clearly visible in the final ensemble of the aLM-EnRML, but seems to be better mitigated in the final ensemble of the RLM-MAC.

The top panels of Figure 13 show the box plots of RMSEs (in permeability values) of ensemble models with respect to the reference model at different iteration steps in the aLM-EnRML (left) and RLM-MAC (right), and bottom ones indicate the corresponding spreads in the ensemble. Similar to the situation in Figure 5, oscillations of the RMSEs (e.g., in terms of the medians of the box plots) are observed in both iES before the RMSE values enter certain plateaus. In this particular case, the final RMSEs of the RLM-MAC tend to be lower than those of the aLM-EnRML. On the other hand, the ensemble spread of the aLM-EnRML tends to decrease with the iteration, while that of the RLM-MAC exhibits slight U-turn behaviour twice before it enters a plateau. The final ensemble spread of the RLM-MAC is higher than that of the aLM-EnRML, and both are underestimated in comparison with the corresponding RMSEs.

Figure 14 shows the normalized differences $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right)$ at different iteration steps, calculated based on the ensemble models of the RLM-MAC. Compared to Figure 6, it is seen that the normalized differences remain relatively small over the iterations, which may be because in this case the phase flows are non-turbulent, and they tend not to amplify the differences among the ensemble models (in contrast to the turbulence nature in the L96 model of the previous subsection).

History matching study in the Brugge field case

Here we compare the history matching performance of the aLM-EnRML and RLM-MAC in the Brugge benchmark case study [31]. The simulation model of the Brugge field case has 60048 ($139 \times 48 \times 9$) gridblocks, with 44550 of them being active. In the case study there are 20 producers and 10 water injectors. The dataset we used is from TNO [31] and contains 20 years' production data⁸. The first decade's data are collected at 127 time instants, while the second decade's at 185 time instants. Following [3], we use the production data at 20 out of 127 time instants in the first decade for history matching, and the rest of the 20 years' production data for out-of-sample cross validation, including, for example, predicting the oil production rates during the second decade based on the history-matched models.

In the course of history matching, the producers and injectors are controlled by the liquid rate (LRAT) target. The production data consist of the historical oil production rates (WOPRH) and water cuts (WWCTH) at 20 producers, and the historical bottom hole pressure (WBHPH) at all 30 wells. Therefore the total number of production data is 1400. The standard deviations for WOPRH, WWCTH and WBHPH are 100 stb/day, 0.05 and 50 psi, respectively. The model variables to be updated include porosity (PORO), permeability (PERMX, PERMY, PERMZ) and net-to-gross (NTG) ratio at all active gridblocks. Consequently the total number of parameters is 222,750. In the aLM-EnRML we use the 104 model realizations from the initial ensemble provided by TNO [31], while in the RLM-MAC we only use 103 of them, after dropping an ensemble member at random (so that the aLM-EnRML and RLM-MAC have the same number of forward runs at each iteration). In both iES, the parameter

⁸ The second decade's production data are provided by TNO based on IRIS' proposed production optimization strategy for that period, see, for example, [31].

rule is $\gamma^i = \alpha^i \text{trace}(\mathbf{S}_d^i (\mathbf{S}_d^i)^T) / N_e$, with the maximum number of iterations being 20, and the coefficient $\beta_u = 0.5$. In all experiment runs we use ECLIPSE[®] [1] for reservoir simulation.

Figure 15 indicates the distributions of PERMX on layers 1 and 2. The top images are from a realization of the reservoir models in the initial ensemble, the middle ones from the corresponding reservoir model updated by the aLM-EnRML, and the bottom ones from the corresponding model obtained by the RLM-MAC. Although the final data mismatch values of the aLM-EnRML and RLM-MAC are close to each other (see Figure 16), there do not seem to be substantial similarities between the resulting PERMX images in Figure 15. This may be due to the difference of both iES in their ways to approximate the gradient of the objective function, hence the difference in the subsequent search directions, as we have discussed previously.

Figure 16 shows the box plots of data mismatch of the aLM-EnRML (left) and RLM-MAC (right) as the function of iteration steps. The aLM-EnRML stops at the 15th iteration step due to the stopping condition on the relative change of data mismatch (less than 0.01%), and the RLM-MAC stops at the maximum iteration step. The aLM-EnRML and RLM-MAC have close performance again, in terms of the convergence speed and the final data mismatch value. In this case, the final data mismatch of the aLM-EnRML is 869.90 ± 23.38 , while that of the RLM-MAC is 928.96 ± 25.76 , about 7% higher in the average data mismatch value.

Figure 17 depicts the history matching profiles of oil production rates (WOPR) at producers BR-P-9 (top row), BR-P-13 (middle row) and BR-P-19 (bottom row) in the first 10 years, with respect to the initial ensemble of reservoir models (1st column), the history matched reservoir models of the aLM-EnRML using production data at 20 time instants (2nd column), and the counterpart history matched reservoir models of the RLM-MAC (3rd column), respectively. In the pictures, the red dots represent the historical WOPRH data at the producers, and the blue curves are the forecasts with respect to the initial ensemble (1st column), and the history matching profiles of both smoothers (2nd and 3rd columns). Consistent with the results in Figure 16, the aLM-EnRML and RLM-MAC exhibit close performance in matching the WOPRH data. Similar results are also obtained for history matching the WWCTH data at BR-P-9, BR-P-13 and BR-P-19, as can be seen in Figure 18.

Figure 19 also shows the normalized differences $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right)$ at different iteration steps, calculated based on the ensemble models of the RLM-MAC, in the Brugge field case. Similar to Figure 14, the normalized differences remain relatively small over the iterations, in contrast to the corresponding results in the L96 model (see Figure 6). This may be also because in this case the reservoir fluid dynamics is non-turbulent, and tends to be less nonlinear than that in the L96 model.

Next we present some cross validation results. Figure 20 shows the box plots of the data mismatch values in 20 years, with respect to the reservoir models from the initial ensemble (left), the final ensemble of the aLM-EnRML using production data at 20 time instants in the first 10 years (middle), and the corresponding final ensemble of the RLM-MAC (right). In this case, the data mismatch values with respect to the initial ensemble and the final ensembles of the aLM-EnRML and RLM-MAC are $3.48 \times 10^5 \pm 2.46 \times 10^5$, 4707.93 ± 357.83 and 4540.31 ± 285.34 , respectively. Therefore, the reservoir models history matched by both iES lead to substantially lower data mismatch than those of the initial ensemble. The data mismatch values of the aLM-EnRML and RLM-MAC are close to each other again, with that of the RLM-MAC being slightly lower.

Figure 21 depicts the WOPR profiles at producers BR-P-9 (top row), BR-P-13 (middle row) and BR-P-19 (bottom row) in 20 years, with respect to the reservoir models from the initial ensemble (1st column), the final ensemble of the aLM-EnRML (2nd column), and the corresponding final ensemble of the RLM-MAC (3rd column), respectively. In the pictures, the red dots represent the historical WOPRH data at the producers, and the blue curves are the forecasts with respect to the initial ensemble and the final ensembles obtained by both smoothers. The pictures in the 2nd and 3rd columns also contain some vertical dashed lines, which are used to separate the production periods between the first and second decades. In this case, the aLM-EnRML and RLM-MAC again exhibit close performance in cross-validating the WOPRH data at BR-P-9. However, in the second decade, the aLM-EnRML predicts the WOPRH data at BR-P-19 better than the RLM-MAC does, which may be related to the fact that the WWCTH data at BR-P-19 is better predicted by the aLM-EnRML (see Figure 22). The situation is reverted for the prediction of the WOPRH data at BR-P-13 in the second decade, and the RLM-MAC

outperforms the aLM-EnRML instead. This may be because the WWCTH data at BR-P-13 is better predicted by the RLM-MAC instead, as can be seen in Figure 22.

Conclusion

In this work we showed that an iteration formula similar to those used in the ensemble smoother with multiple data assimilation (ES-MDA) and the approximate Levenberg-Marquardt ensemble randomized maximum likelihood (aLM-EnRML) method can be derived from the regularized Levenberg-Marquardt (RLM) algorithm in inverse problems theory, when history matching is recast as a minimum-average-cost (MAC) problem.

Specifically, to reduce the computational cost in solving the MAC problem, a simple strategy is to approximate the nonlinear simulator, for all ensemble models, through a first order Taylor expansion around a single common point, e.g., the ensemble mean at the previous iteration step as considered in the current study. In addition, to avoid directly evaluating the resulting (common) Jacobian matrix, one may follow the convention in the ensemble Kalman filter (EnKF) to approximate the product between the Jacobian matrix and the model square root matrix by the corresponding data square root matrix. In this way, the resulting iteration formula is similar to those in the ES-MDA and aLM-EnRML, but differs in the way of constructing the data square root matrix: In the ES-MDA and aLM-EnRML, the simulated data center around their sample mean, while in the RLM-MAC, they center around the simulated data of the mean of the ensemble models. In general, it is expected that one may solve the MAC problem in a more accurate way – but possibly with a higher computational cost – by evaluating the gradients at multiple common points.

To examine the effect of different ways in constructing the data square root matrix, we compare the performance of the aLM-EnRML and RLM-MAC in three numerical examples. In the first example we consider a strongly nonlinear system (the Lorenz 96 model). In this particular case the aLM-EnRML and RLM-MAC exhibit more diverged behaviour and the RLM-MAC tends to perform better than the aLM-EnRML. In the second and third examples we apply the aLM-EnRML and RLM-MAC to a reservoir facies estimation problem and the history matching problem in the Brugge field case, respectively. In

both cases it appears that the aLM-EnRML and RLM-MAC exhibit close performance due to weaker nonlinearity in the reservoir fluid dynamics.

Nomenclature

\mathbf{d}	=	data
\mathbf{d}^o	=	observation data
\mathbf{f}	=	function
\mathbf{g}	=	reservoir simulator
m	=	dimension of reservoir model
\mathbf{m}	=	reservoir model
\mathbf{m}_c	=	common reservoir model in the minimum-average-cost problem
$\bar{\mathbf{m}}$	=	ensemble-average reservoir model
N_e	=	ensemble size
p	=	number of observation data
p_k	=	probability density function (pdf) at the k th iteration step
r	=	scalar coefficient larger than 1
t	=	unit-less sudo-time variable in the Lorenz 96 model
\mathbf{v}	=	generic random vector
$\hat{\mathbf{v}}$	=	estimation of \mathbf{v}
\mathbf{v}^{tr}	=	true realization value of the random vector \mathbf{v}
x	=	state variable in the Lorenz 96 model
\mathbf{y}	=	random vector that represents observations
\mathbf{A}	=	generic matrix
\mathbf{C}_d	=	observation error covariance matrix
\mathbf{C}_m	=	model error covariance matrix as in the EnKF
$\tilde{\mathbf{C}}_m$	=	modified model error covariance matrix in the RLM-MAC
F	=	unit-less driving force in the Lorenz 96 model
\mathcal{F}	=	functional
\mathcal{G}	=	inverse functional of \mathcal{F}
\mathbf{G}	=	Jacobian of reservoir simulator
\mathcal{G}^C	=	composite functional
I	=	number of iterations in ES-MDA

\mathbf{I}_p = identity matrix of size p

\mathbf{M} = ensemble of reservoir models

\mathcal{O} = objective function

$\bar{\mathcal{O}}$ = expectation of \mathcal{O}

\mathbf{S}_d = data square root matrix as in the EnKF

$\tilde{\mathbf{S}}_d$ = modified data square root matrix in the RLM-MAC

\mathbf{S}_m = model square root matrix

Greek symbols

α = adaptive coefficient in the RLM-MAC that determines γ

δ = Dirac delta function

γ = adaptive coefficient in iterative ensemble smoothers

σ^2 = sample variances of the ensemble of reservoir models

ρ = tapering factor of γ

Δm = model change

Subscript

a = analysis

b = background

d = data

e = ensemble

j = index of ensemble member

k = iteration index

m = model

s = index of sample variances of the ensemble of reservoir models

Superscript

b = background

i = iteration index

o = observation

tr = truth

Acknowledgements We would like to thank Drs. Yan Chen (Total), Randi Valestrand (IRIS) and three anonymous reviewers for their valuable helps, comments and suggestions, which have significantly improved the work. We also acknowledge the IRIS/CIPR cooperative research project “Integrated Workflow and Realistic Geology” which is funded by industry partners ConocoPhillips, Eni, Petrobras, Statoil, and Total, as well as the Research Council of Norway (PETROMAKS) for financial support, and thank Schlumberger for ECLIPSE academic licenses and the developers of SGeSM and MATLAB level set toolbox (Ian M. Mitchell) for providing their softwares.

A Characterizing the RLM-MAC from the point of view of an expectation-maximization algorithm

Following a recent working paper [20], the RLM-MAC algorithms can also be described in analogy to the expectation-maximization (EM) algorithm [5], which is an iterative algorithm that consists of an expectation step (E-step) and an maximization step (M-step) at each iteration. At the E-step, one constructs an expectation-type cost function that depends on certain unknown quantities-of-interest (QOI) and their prior knowledge, while at the subsequent M-step, one solves a maximization problem to obtain an updated estimation of the QOI. The updated QOI are then passed to the next iteration as the new prior knowledge of the corresponding E-step, and so on.

The cost function in the RLM-MAC can be considered as a Monte-Carlo approximation to the following expectation-type cost function at the E-step of the k th iteration:

$$\bar{\mathcal{O}}_k = \int \mathcal{O}_k(\mathbf{m}_a^k, \mathbf{m}_b^k, \tilde{\mathbf{y}}) p_k(\mathbf{m}_a^k, \mathbf{m}_b^k, \tilde{\mathbf{y}}|\mathbf{y}) d\mathbf{m}_b^k d\mathbf{m}_a^k d\tilde{\mathbf{y}}, \quad (\text{A.1})$$

where \mathbf{m}_b^k is the prior knowledge of the QOI, \mathbf{m}_a^k the QOI to be updated, \mathbf{y} the real observation, and $\tilde{\mathbf{y}}$ a perturbation of \mathbf{y} ; $\mathcal{O}_k(\mathbf{m}_a^k, \mathbf{m}_b^k, \tilde{\mathbf{y}})$ is a chosen function (e.g., as the quadratic function in the RLM-MAC) and may depend on \mathbf{m}_b^k , \mathbf{m}_a^k and $\tilde{\mathbf{y}}$, and $\bar{\mathcal{O}}_k$ is the expectation of $\mathcal{O}_k(\mathbf{m}_a^k, \mathbf{m}_b^k, \tilde{\mathbf{y}})$ with respect to the (conditional) joint pdf $p_k(\mathbf{m}_a^k, \mathbf{m}_b^k, \tilde{\mathbf{y}}|\mathbf{y})$. In the context of data assimilation, the updated QOI \mathbf{m}_a^k often depend on both \mathbf{m}_b^k and $\tilde{\mathbf{y}}$, so that one has $\mathbf{m}_a^k = \mathcal{F}_k(\mathbf{m}_b^k, \tilde{\mathbf{y}})$, where \mathcal{F}_k is an unknown deterministic map (or functional), and is determined at the subsequent “M-step” under a certain criterion. In the RLM-MAC, \mathcal{F}_k are determined by solving the following functional optimization problem

$$\operatorname{argmax}_{\mathcal{F}_k} \bar{\mathcal{O}}_k. \quad (\text{A.2})$$

Note that \mathcal{F}_k may also be chosen from other points of views. For instance, they may be constructed in a way such that the conditional pdf $p_k(\mathbf{m}_a^k|\mathbf{y})$ is consistent with Bayes’ rule, in the sense of satisfying

$$p_k(\mathbf{m}_a^k|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{m}_a^k)p_0(\mathbf{m}_a^k)}{p(\mathbf{y})}. \quad (\text{A.3})$$

On the other hand, note that $\mathbf{m}_b^k = \mathbf{m}_a^{k-1}$. Through recursion, one has $\mathbf{m}_a^k = \mathcal{F}_k(\mathbf{m}_b^k, \tilde{\mathbf{y}}) = \mathcal{F}_k(\mathcal{F}_{k-1}(\mathbf{m}_b^{k-1}, \tilde{\mathbf{y}}), \tilde{\mathbf{y}}) = \dots$. Let $\mathbf{m}_a^k \equiv \mathcal{F}_k^C(\mathbf{m}_b^0, \tilde{\mathbf{y}})$, where \mathcal{F}_k^C is the composition of the functionals \mathcal{F}_i , then by the definition of pdf of transformed

random variables we have the relation between $p_0(\mathbf{m}_b^0, \tilde{\mathbf{y}}|\mathbf{y})$ and $p_k(\mathbf{m}_a^k|\mathbf{y})$ given by

$$\begin{aligned} p_k(\mathbf{m}_a^k|\mathbf{y}) &= \frac{\partial}{\partial \mathbf{m}_a^k} \int_{\mathcal{F}_k^C(\mathbf{m}_b^0, \tilde{\mathbf{y}}) \leq \mathbf{m}_a^k} p_0(\mathbf{m}_b^0, \tilde{\mathbf{y}}|\mathbf{y}) d\mathbf{m}_b^0 d\tilde{\mathbf{y}} \\ &= \frac{\partial}{\partial \mathbf{m}_a^k} \int_{\mathcal{F}_k^C(\mathbf{m}_b^0, \tilde{\mathbf{y}}) \leq \mathbf{m}_a^k} p_0(\mathbf{m}_b^0) p(\tilde{\mathbf{y}}|\mathbf{y}) d\mathbf{m}_b^0 d\tilde{\mathbf{y}}, \end{aligned} \quad (\text{A.4})$$

where $p(\tilde{\mathbf{y}}|\mathbf{y})$ is the pdf of the perturbed observation $\tilde{\mathbf{y}}$, given the real observation \mathbf{y} . Note that in the second line of Eq. (A.4), we have assumed that the initial prior \mathbf{m}_b^0 is independent of observation. Combining Eqs. (A.4) and (A.3), we require the following identity

$$\frac{\partial}{\partial \mathbf{m}_a^k} \int_{\mathcal{F}_k^C(\mathbf{m}_b^0, \tilde{\mathbf{y}}) \leq \mathbf{m}_a^k} p_0(\mathbf{m}_b^0) p(\tilde{\mathbf{y}}|\mathbf{y}) d\mathbf{m}_b^0 d\tilde{\mathbf{y}} = \frac{p(\mathbf{y}|\mathbf{m}_a^k) p_0(\mathbf{m}_a^k)}{p(\mathbf{y})} \quad (\text{A.5})$$

to hold in order for $p_k(\mathbf{m}_a^k|\mathbf{y})$ to satisfy Bayes' rule.

Eq. (A.5) suggests that the choice of \mathcal{F}_i depends on both the prior pdf $p_0(\bullet)$ and the likelihood $p(\mathbf{y}|\bullet)$. As a side remark, we also note that Eq. (A.5) only imposes a constraint on the compositional functional \mathcal{G}_k^C , but does not specify how \mathcal{F}_i should be chosen at individual iteration steps. This freedom allows one to incorporate additional criteria (e.g., in terms of solving optimization problems in (A.2)) at some of the iteration steps, subject to the condition in Eq. (A.5).

In linear systems with Gaussian model and observation errors, [6] provides proof and examples on how to construct the functionals \mathcal{F}_i to make the resulting $p_k(\mathbf{m}_a^k|\mathbf{y})$ consistent with Bayes' rule. In general situations with nonlinearity and/or non-Gaussianity, the constructions of \mathcal{F}_i become more complicated and are beyond the scope of the current study.

B Derivation of the iteration formula in RLM-MAC

By inserting Eq. (18) into (17), one has the following approximate cost function

$$\begin{aligned} \tilde{J}^{i+1} &= \frac{1}{N_e} \sum_{j=1}^{N_e} \left[\left(\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_c^i) - \mathbf{G}_c^i (\mathbf{m}_j^{i+1} - \mathbf{m}_c^i) \right)^T \mathbf{C}_d^{-1} \left(\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_c^i) - \mathbf{G}_c^i (\mathbf{m}_j^{i+1} - \mathbf{m}_c^i) \right) \right. \\ &\quad \left. + \gamma^i (\mathbf{m}_j^{i+1} - \mathbf{m}_j^i)^T (\tilde{\mathbf{C}}_m^i)^{-1} (\mathbf{m}_j^{i+1} - \mathbf{m}_j^i) \right]. \end{aligned} \quad (\text{B.1})$$

A necessary condition for \mathbf{m}_j^{i+1} ($j = 1, 2, \dots, N_e$) being the solution of the MAC problem is thus $\partial \tilde{J}^{i+1} / \partial \mathbf{m}_j^{i+1} = 0$, which leads to the following equation:

$$-2 (\mathbf{G}_c^i)^T \mathbf{C}_d^{-1} \left(\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_c^i) - \mathbf{G}_c^i (\mathbf{m}_j^{i+1} - \mathbf{m}_c^i) \right) + 2 \gamma^i (\tilde{\mathbf{C}}_m^i)^{-1} (\mathbf{m}_j^{i+1} - \mathbf{m}_j^i) = 0. \quad (\text{B.2})$$

With some algebraic operations, the above equation becomes

$$\left[(\mathbf{G}_c^i)^T \mathbf{C}_d^{-1} \mathbf{G}_c^i + \gamma^i (\tilde{\mathbf{C}}_m^i)^{-1} \right] (\mathbf{m}_j^{i+1} - \mathbf{m}_j^i) = (\mathbf{G}_c^i)^T \mathbf{C}_d^{-1} [\mathbf{d}_j^o - \mathbf{g}(\mathbf{m}_c^i) - \mathbf{G}_c^i (\mathbf{m}_j^i - \mathbf{m}_c^i)]. \quad (\text{B.3})$$

By the Sherman-Morrison-Woodbury formula [32], one has

$$\left[(\mathbf{G}_c^i)^T \mathbf{C}_d^{-1} \mathbf{G}_c^i + \gamma^i (\tilde{\mathbf{C}}_m^i)^{-1} \right]^{-1} (\mathbf{G}_c^i)^T \mathbf{C}_d^{-1} = \tilde{\mathbf{C}}_m^i (\mathbf{G}_c^i)^T \left[\mathbf{G}_c^i \tilde{\mathbf{C}}_m^i (\mathbf{G}_c^i)^T + \gamma^i \mathbf{C}_d \right]^{-1}.$$

Inserting the above identity into Eq. (B.3) and letting $\tilde{\mathbf{C}}_m^i = \tilde{\mathbf{S}}_m^i (\tilde{\mathbf{S}}_m^i)^T$, one obtains Eq. (19).

References

1. Schlumberger. “ECLIPSE 2010 Reservoir Engineering Software”
2. Aanonsen, S., Nævdal, G., Oliver, D., Reynolds, A., Vallès, B.: The ensemble Kalman filter in reservoir engineering: a review. *SPE Journal* **14**(3), 393–412 (2009)
3. Chen, Y., Oliver, D.: Levenberg-Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification. *Computational Geosciences* **17**, 689–703 (2013)
4. Chen, Y., Oliver, D.S.: Ensemble randomized maximum likelihood method as an iterative ensemble smoother. *Mathematical Geosciences* **44**, 1–26 (2012)
5. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* **39**, 1–38 (1977)
6. Emerick, A.A., Reynolds, A.C.: Ensemble smoother with multiple data assimilation. *Computers & Geosciences* **55**, 3–15 (2012)
7. Engl, H.W., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems*. Springer (2000)
8. Evensen, G.: *Data Assimilation: The Ensemble Kalman Filter*. Springer (2006)
9. Evensen, G., van Leeuwen, P.J.: An ensemble Kalman smoother for nonlinear dynamics. *Mon. Wea. Rev.* **128**, 1852–1867 (2000)
10. Gu, Y., Oliver, D.: An iterative ensemble Kalman filter for multiphase fluid flow data assimilation. *SPE Journal* **12**, 438–446 (2007)
11. Hanke, M.: A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse problems* **13**(1), 79–95 (1997)
12. Iglesias, M.A.: Iterative regularization for ensemble data assimilation in reservoir models. *Computational Geosciences*, in review
13. Iglesias, M.A., Dawson, C.: The regularizing Levenberg-Marquardt scheme for history matching of petroleum reservoirs. *Computational Geosciences* **17**, 1033–1053 (2013)
14. Jin, Q.: On a regularized Levenberg–Marquardt method for solving nonlinear inverse problems. *Numerische Mathematik* **115**, 229–259 (2010)
15. Kaltenbacher, B., Neubauer, A., Scherzer, O.: *Iterative regularization methods for nonlinear ill-posed problems*. Walter de Gruyter (2008)
16. Li, G., Reynolds, A.: Iterative ensemble Kalman filters for data assimilation. *SPE Journal* **14**, 496–505 (2009)
17. Lorentzen, R., Flornes, K., Nævdal, G.: History matching channelized reservoirs using the ensemble Kalman filter. *SPE Journal* **17**, 137–151 (2012)
18. Lorentzen, R., Nævdal, G.: An iterative ensemble Kalman filter. *IEEE Transactions on Automatic Control* **56**, 1990–1995 (2011)
19. Lorenz, E.N., Emanuel, K.A.: Optimal sites for supplementary weather observations: Simulation with a small model. *J. Atmos. Sci.* **55**, 399–414 (1998)

-
20. Luo, X.: Optimization under uncertainty: A unified framework for a class of ensemble data assimilation algorithms. Under revision (2015)
 21. Luo, X., Chen, Y., Valestrand, R., Stordal, A., Lorentzen, R., Nævdal, G.: On an alternative implementation of the iterative ensemble smoother and its application to reservoir facies estimation. In: ECMOR XIV-14th European conference on the mathematics of oil recovery. Catania, Sicily, Italy (2014)
 22. Luo, X., Hoteit, H.: Ensemble Kalman filtering with residual nudging: an extension to the state estimation problems with nonlinear observations. *Mon. Wea. Rev.* **142**, 3696–3712 (2014)
 23. Luo, X., Hoteit, I.: Ensemble Kalman filtering with residual nudging. *Tellus A* **64**, 17,130, open access (2012). DOI 10.3402/tellusa.v64i0.17130
 24. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics* **11**(2), 431–441 (1963)
 25. Mitchell, I.M.: The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing* **35**, 300–329 (2008)
 26. Nævdal, G., Mannseth, T., Vefring, E.H.: Near-well reservoir monitoring through ensemble Kalman filter. In: SPE/DOE Improved Oil Recovery Symposium, pp. 13–17. Tulsa, Oklahoma, USA (2002). SPE-75235-MS
 27. Nævdal, G., Thulin, K., Skaug, H.J., Aanonsen, S.I.: Quantifying Monte Carlo uncertainty in the ensemble Kalman filter. *SPE Journal* **16**, 172–182 (2011)
 28. Nocedal, J., Wright, S.J.: Numerical optimization, 2nd edn. Springer (2006)
 29. Oliver, D.S.: Minimization for conditional simulation: Relationship to optimal transport. *Journal of Computational Physics* **265**, 1–15 (2014)
 30. Oliver, D.S., Reynolds, A.C., Liu, N.: Inverse Theory for Petroleum Reservoir Characterization and History Matching. Cambridge University Press (2008)
 31. Peters, L., Arts, R., Brouwer, G., Geel, C., Cullick, S., Lorentzen, R.J., Chen, Y., Dunlop, N., Vossepoel, F.C., Xu, R., et al.: Results of the Brugge benchmark study for flooding optimization and history matching. *SPE Reservoir Evaluation & Engineering* **13**, 391–405 (2010)
 32. Sherman, J., Morrison, W.J.: Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics* **21**, 124–127 (1950)
 33. Skjervheim, J.A., Evensen, G.: An ensemble smoother for assisted history matching. In: SPE Reservoir Simulation Symposium. The Woodlands, Texas, USA (2011). SPE-141929-MS
 34. Stordal, A.: Iterative Bayesian inversion with Gaussian mixtures: finite sample implementation and large sample asymptotics. *Computational Geosciences*, in press (2014). DOI 10.1007/s10596-014-9444-9
 35. Strebelle, S.: Conditional simulation of complex geological structures using multiple-point statistics. *Mathematical Geology* **34**, 1–21 (2002)
 36. Tavakoli, R., Reynolds, A.C., et al.: History matching with parametrization based on the SVD of a dimensionless sensitivity matrix. *SPE Journal* **15**, 495–508 (2010)

-
37. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on* **1**, 67–82 (1997)

List of Figures

- 1 Histograms of (a) data mismatch and (b) RMSEs of the aLM-EnRML and RLM-MAC obtained from 100 repetitions of the experiment in the L96 model. For visualization, the horizontal axes of Figure 1(a) are in the logarithmic scale. In Figure 1(b), the RMSEs are calculated based on the differences between the trajectory generated by the true initial state (in 10 days) and those generated by the estimated ones. 36
- 2 Pie charts of the data mismatch of the aLM-EnRML (left) and RLM-MAC (right) with respect to the results in Fig. 1. 37
- 3 Pie charts of the RMSEs of the aLM-EnRML (left) and RLM-MAC (right) with respect to the results in Fig. 1. 38
- 4 Box plots of the data mismatch of the aLM-EnRML (left) and the RLM-MAC (right) at different iteration steps in one run in the L96 model. The maximum number of iterations for the RLM-MAC is 100, while that for the aLM-EnRML is increased to 200 and the aLM-EnRML converges after 120 iteration steps. For visualization, the vertical axes are in the logarithmic scale, and the results are plotted at iteration steps 0, 5, and every 5 iteration steps thereafter, along the horizontal axes. In each box plot, the horizontal line (in red) inside the box denotes the median; the top and bottom of the box represent the 75th and 25th percentiles, respectively; the whiskers indicate the ranges beyond which the data are considered outliers, and the whiskers' positions are determined using the default MATLAB[®] setting (please see the corresponding documentation in MATLAB[®] R2012a), while the outliers themselves are plotted individually as plus signs (in red). 39
- 5 Top panels: Box plots of RMSEs of the ensemble members with respect to the initial conditions at different iteration steps in the aLM-EnRML (left) and RLM-MAC (right). Bottom panels: Ensemble spreads of the aLM-EnRML (left) and RLM-MAC (right) at different iteration steps of the L96 model. 40
- 6 Box plots of the normalized differences $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right)$ at different iteration steps in the L96 model. 41
- 7 Reference facies model. The white dots indicate the locations of injectors (I1 - I8) and producers (P1 - P8). 42
- 8 A few models (members 1 - 3) in the initial ensemble (1st column) and the ensembles of the aLM-EnRML (2nd column) and RLM-MAC (3rd column) at the final iteration steps of the facies estimation problem. . . 43
- 9 Average facies matching scores of (a) the initial ensemble, (b) the final ensemble of the aLM-EnRML and (c) the final ensemble of the RLM-MAC. A higher score means that more models have the correct facies. . . . 44
- 10 Box plots of data mismatch at different iteration steps of the facies estimation problem. Left: aLM-EnRML; Right: RLM-MAC. 45

- 11 History matching of oil production rates at P5 (top row) and P7 (bottom row) of the facies estimation problem, using the initial ensemble (1st column) and the ensembles of the aLM-EnRML (2nd column) and RLM-MAC (3rd column) at the final iteration steps. In each image, the red dots represent the synthetic measurements and the blue curves are the forecasts with respect to the initial ensemble (1st column) and history matching profiles (2nd and 3rd columns). For comparison, the production data of the reference model are also shown in orange. 46
- 12 As in Figure 11, but for history matching of water production rates in the facies estimation problem. 47
- 13 Top panels: Box plots of RMSEs of the ensemble models—with respect to the reference model of the facies estimation problem—at different iteration steps in the aLM-EnRML (left) and RLM-MAC (right). Bottom panels: Ensemble spreads of the aLM-EnRML (left) and RLM-MAC (right) at different iteration steps. 48
- 14 Box plots of the normalized differences $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\bar{\mathbf{m}}_j^i)} \right)$ at different iteration steps of the facies estimation problem. 49
- 15 Distributions of PERMX on layers 1 and 2 from a reservoir model of the initial ensemble (a,b), the corresponding reservoir model updated by the aLM-EnRML (c,d), and the corresponding reservoir model updated by the RLM-MAC (e,f). The dots in the figures indicate the locations of the producers and injectors in the Brugge field. 50
- 16 Box plots of data mismatch of the aLM-EnRML (left) and the RLM-MAC (right) at different iteration steps in the Brugge field case, using production data at 20 out of 127 time instants in the first 10 years. 51
- 17 History matching of oil production rates (WOPR) at BR-P-9 (top), BR-P-13 (middle) and BR-P-19 (bottom) in the Brugge field case, using the initial ensemble (1st column) and the ensembles of the aLM-EnRML (2nd column) and RLM-MAC (3rd column) at the final iteration steps. In each image, the red dots represent the historical WOPRH data and the blue curves are the forecasts with respect to the initial ensemble (1st column), and history matching profiles (2nd and 3rd columns). 52
- 18 As in Figure 17, but for water cut (WWCT) at BR-P-9, BR-P-13 and BR-P-19 in the Brugge field case. 53
- 19 Box plots of the normalized differences $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\bar{\mathbf{m}}_j^i)} \right)$ at different iteration steps in the Brugge field case. 54
- 20 Box plots of data mismatch with respect to the initial ensemble (left), the aLM-EnRML (middle) and the RLM-MAC (right) in the Brugge field case. Here, all the production data in the first 20 years are used to cross-validate the history-matched reservoir models obtained by aLM-EnRML and RLM-MAC using production data at 20 out of 127 time instants in the first 10 years. 55

-
- 21 Cross validation of oil production rates (WOPR) at BR-P-9 (top), BR-P-13 (middle) and BR-P-19 (bottom) in the Brugge field case, using the initial ensemble (1st column) and the ensembles of the aLM-EnRML (2nd column) and RLM-MAC (3rd column) at the final iteration steps. In each image, the red dots represent the historical WOPRH data and the blue curves are the forecasts with respect to the initial ensemble (1st column) and the final ensembles obtained by both iES (2nd and 3rd columns). The vertical dashed lines in the figures of the 2nd and 3rd columns separate the first and second decades. 56
- 22 As in Figure 17, but for water cut (WWCT) at BR-P-9, BR-P-13 and BR-P-19 in the Brugge field case. . . 57

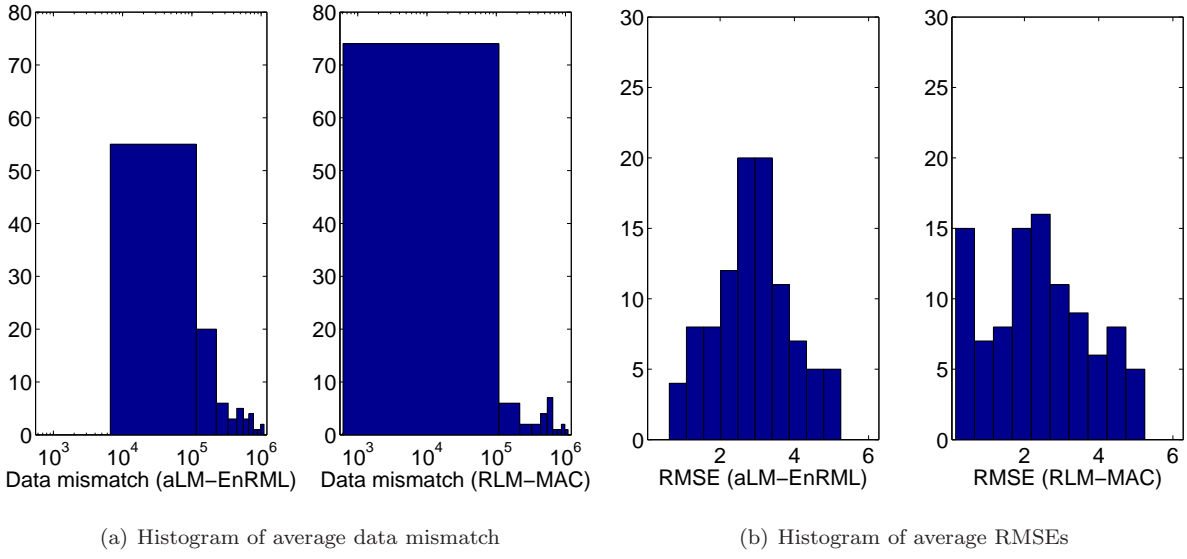


Fig. 1 Histograms of (a) data mismatch and (b) RMSEs of the aLM-EnRML and RLM-MAC obtained from 100 repetitions of the experiment in the L96 model. For visualization, the horizontal axes of Figure 1(a) are in the logarithmic scale. In Figure 1(b), the RMSEs are calculated based on the differences between the trajectory generated by the true initial state (in 10 days) and those generated by the estimated ones.

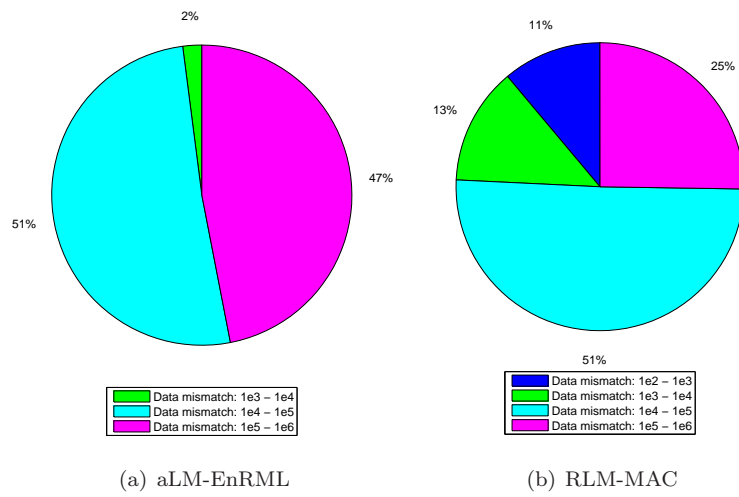


Fig. 2 Pie charts of the data mismatch of the aLM-EnRML (left) and RLM-MAC (right) with respect to the results in Fig. 1.

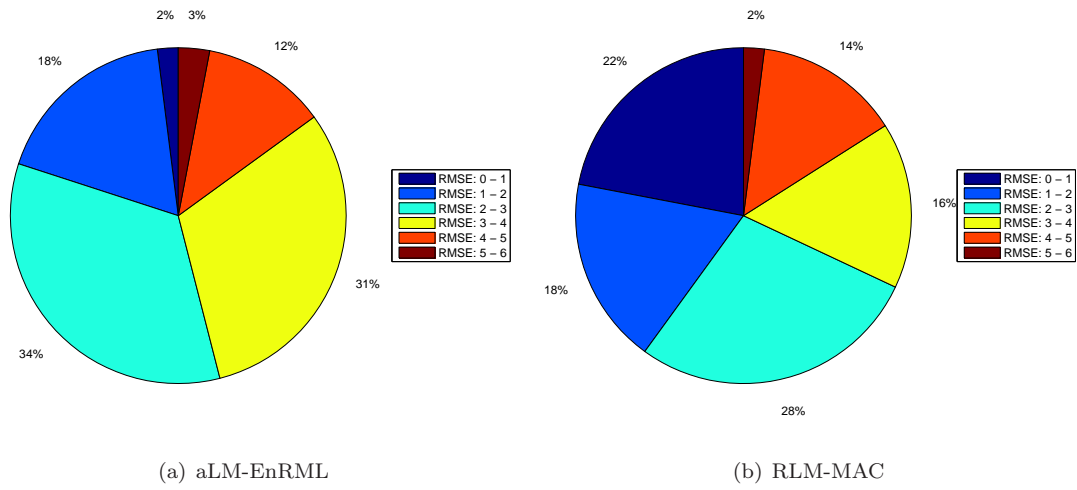


Fig. 3 Pie charts of the RMSEs of the aLM-EnRML (left) and RLM-MAC (right) with respect to the results in Fig. 1.

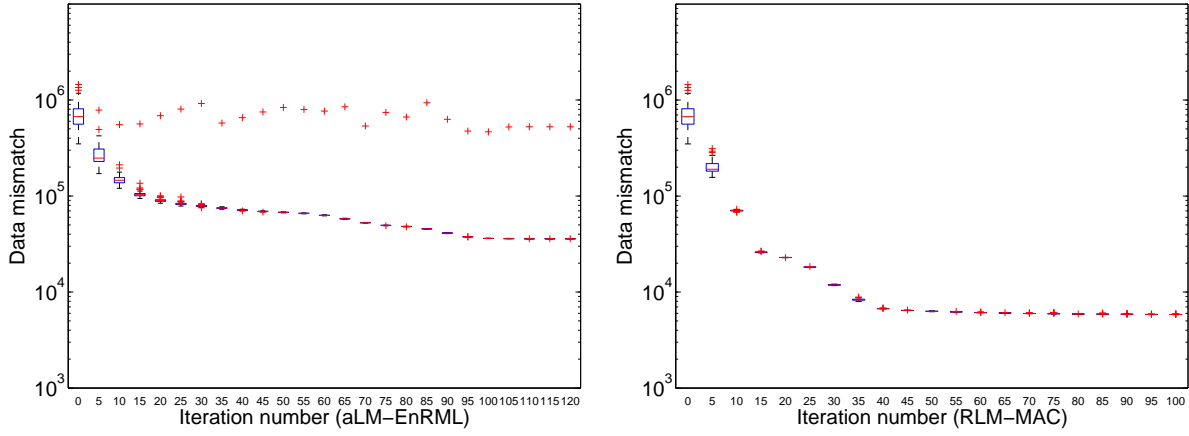


Fig. 4 Box plots of the data mismatch of the aLM-EnRML (left) and the RLM-MAC (right) at different iteration steps in one run in the L96 model. The maximum number of iterations for the RLM-MAC is 100, while that for the aLM-EnRML is increased to 200 and the aLM-EnRML converges after 120 iteration steps. For visualization, the vertical axes are in the logarithmic scale, and the results are plotted at iteration steps 0, 5, and every 5 iteration steps thereafter, along the horizontal axes. In each box plot, the horizontal line (in red) inside the box denotes the median; the top and bottom of the box represent the 75th and 25th percentiles, respectively; the whiskers indicate the ranges beyond which the data are considered outliers, and the whiskers' positions are determined using the default MATLAB[©] setting (please see the corresponding documentation in MATLAB[©] R2012a), while the outliers themselves are plotted individually as plus signs (in red).

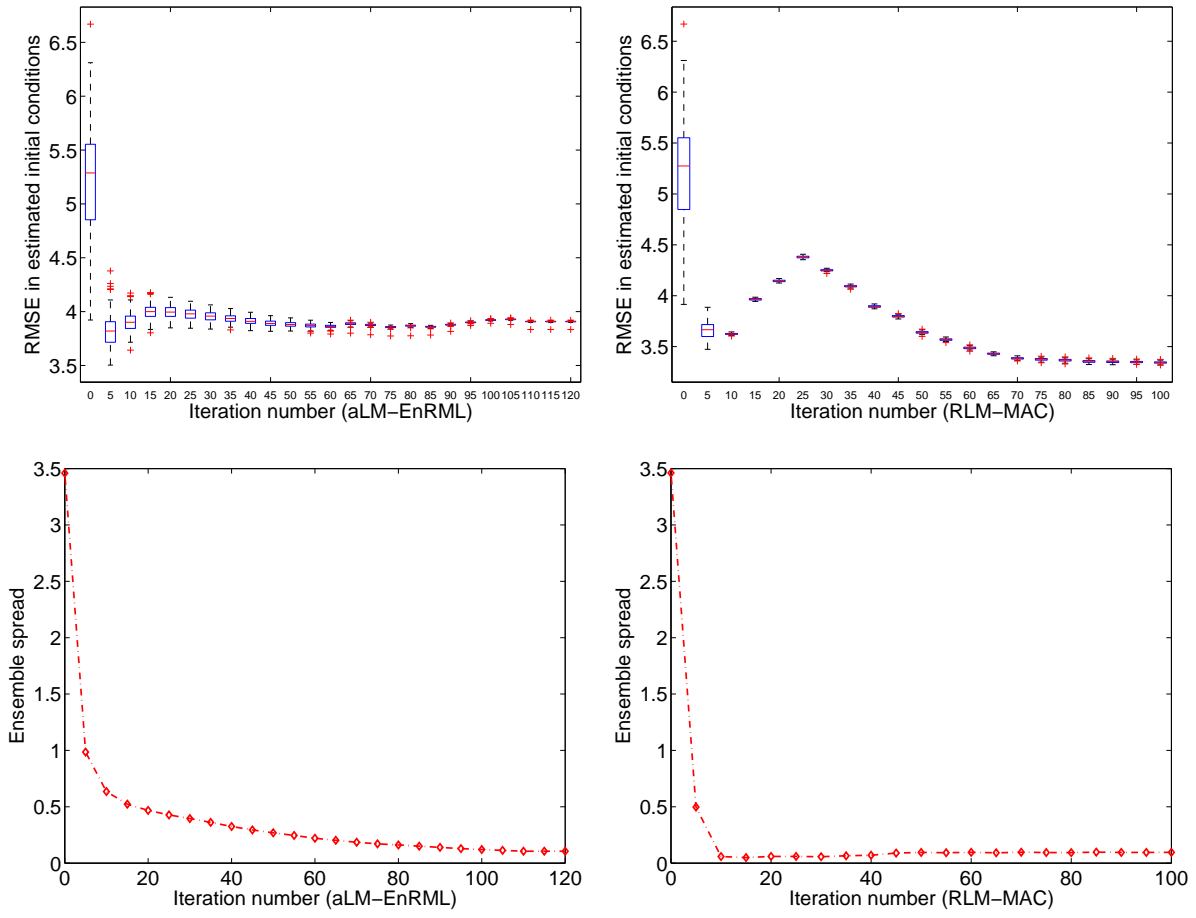


Fig. 5 Top panels: Box plots of RMSEs of the ensemble members with respect to the initial conditions at different iteration steps in the aLM-EnRML (left) and RLM-MAC (right). Bottom panels: Ensemble spreads of the aLM-EnRML (left) and RLM-MAC (right) at different iteration steps of the L96 model.

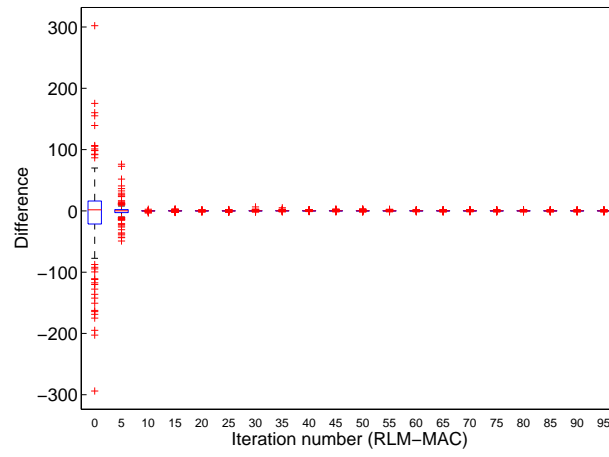


Fig. 6 Box plots of the normalized differences $C_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right)$ at different iteration steps in the L96 model.

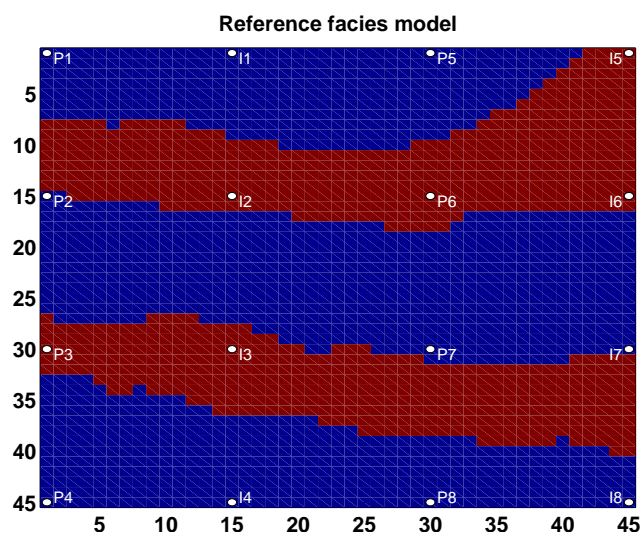


Fig. 7 Reference facies model. The white dots indicate the locations of injectors (I1 - I8) and producers (P1 - P8).

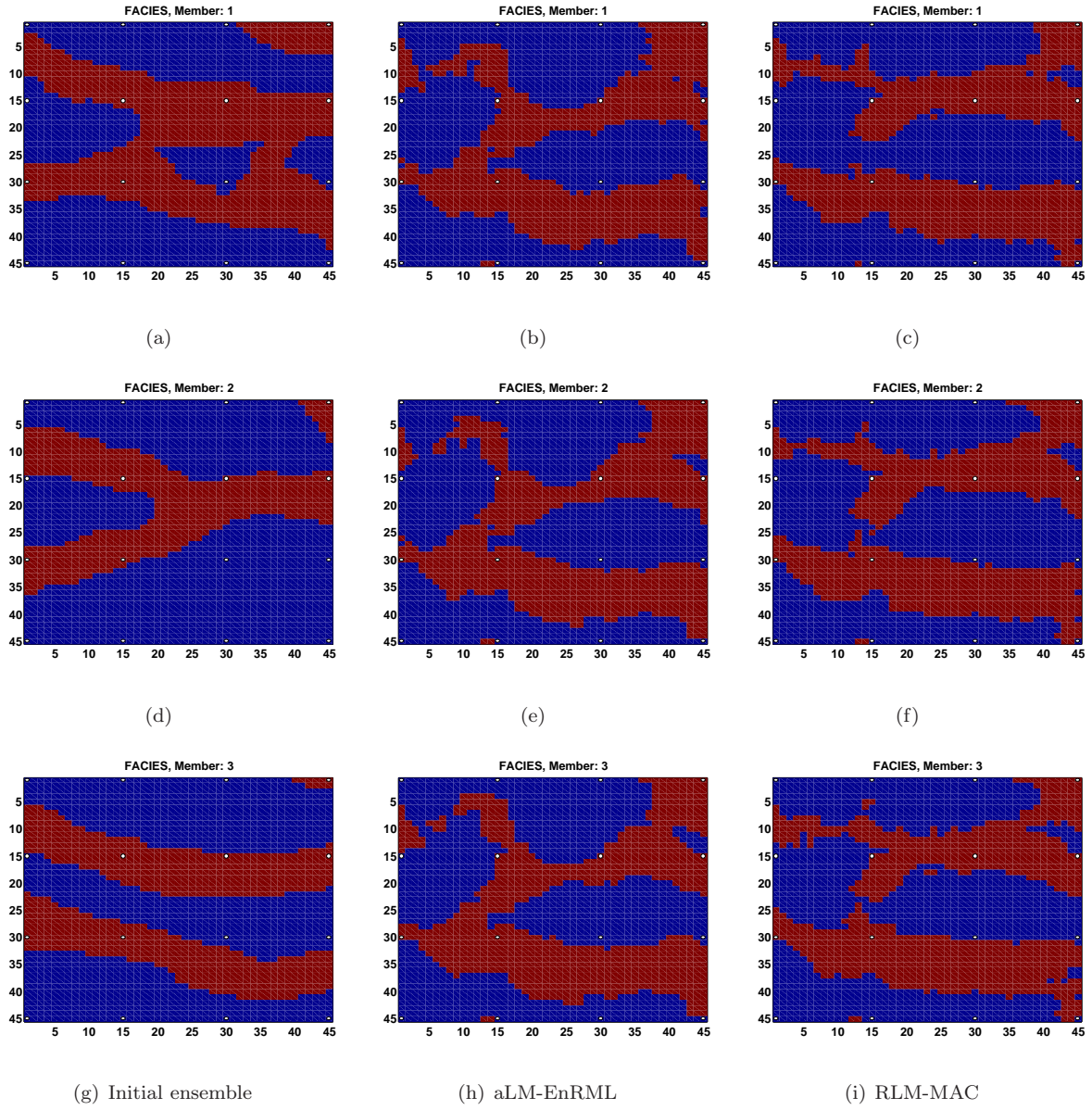


Fig. 8 A few models (members 1 - 3) in the initial ensemble (1st column) and the ensembles of the aLM-EnRML (2nd column) and RLM-MAC (3rd column) at the final iteration steps of the facies estimation problem.

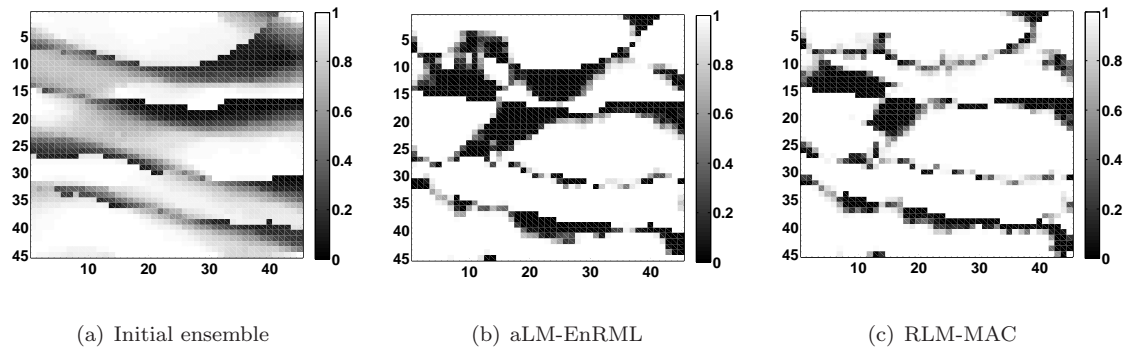


Fig. 9 Average facies matching scores of (a) the initial ensemble, (b) the final ensemble of the aLM-EnRML and (c) the final ensemble of the RLM-MAC. A higher score means that more models have the correct facies.

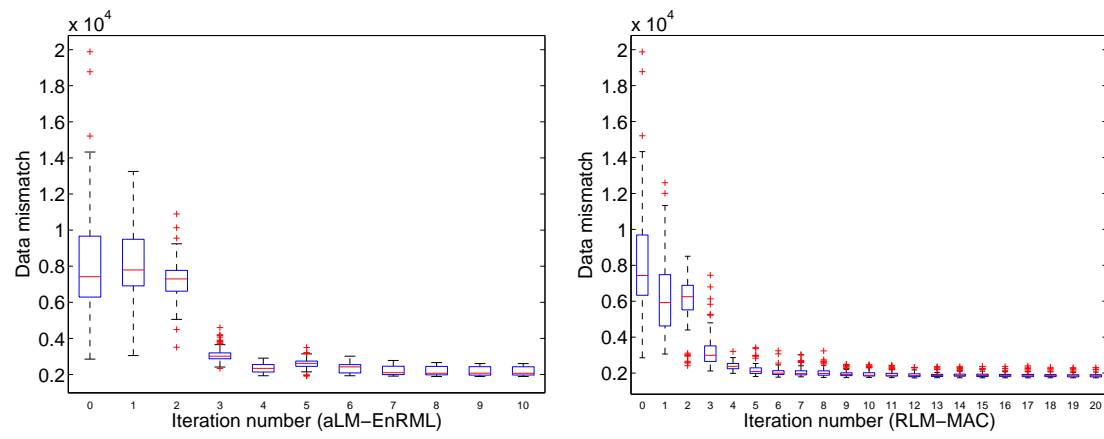


Fig. 10 Box plots of data mismatch at different iteration steps of the facies estimation problem. Left: aLM-EnRML; Right: RLM-MAC.

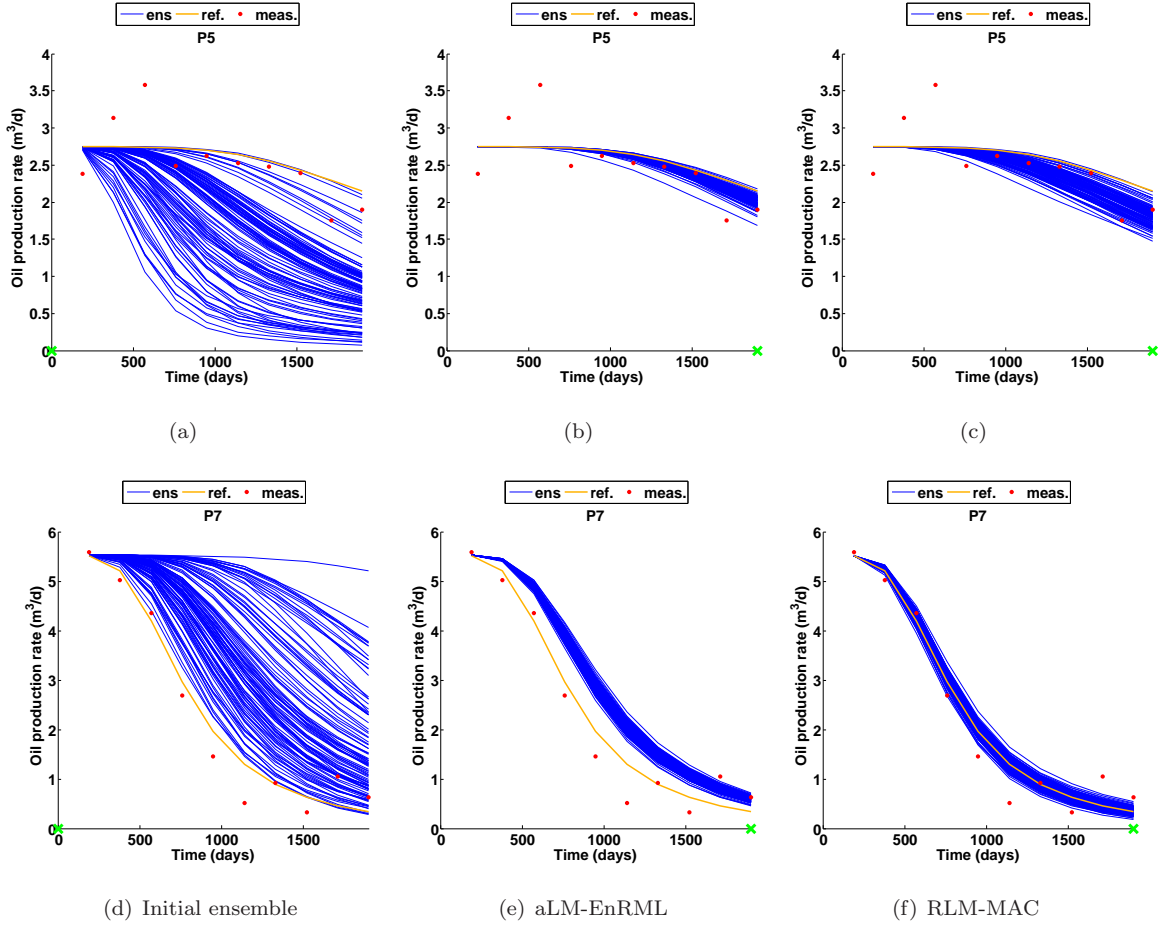


Fig. 11 History matching of oil production rates at P5 (top row) and P7 (bottom row) of the facies estimation problem, using the initial ensemble (1st column) and the ensembles of the aLM-EnRML (2nd column) and RLM-MAC (3rd column) at the final iteration steps. In each image, the red dots represent the synthetic measurements and the blue curves are the forecasts with respect to the initial ensemble (1st column) and history matching profiles (2nd and 3rd columns). For comparison, the production data of the reference model are also shown in orange.

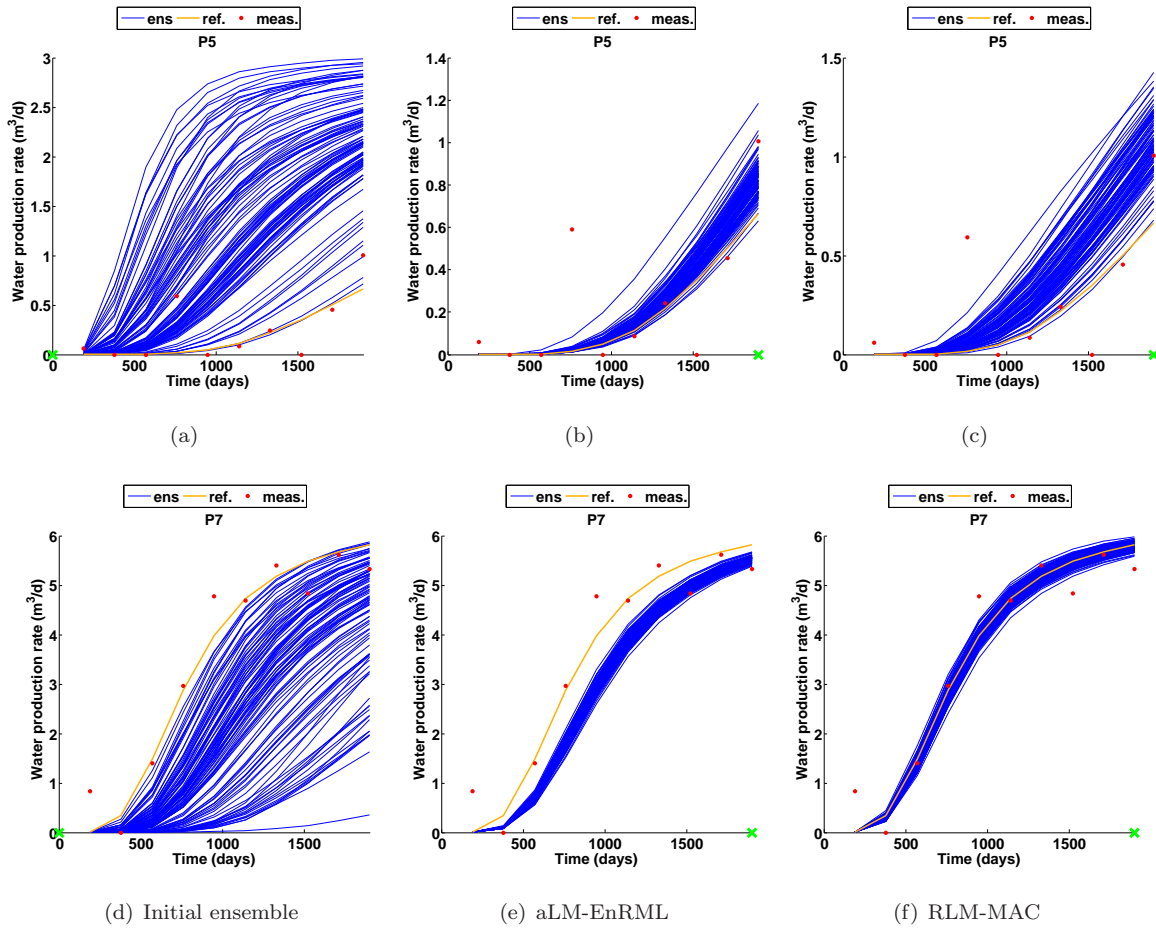


Fig. 12 As in Figure 11, but for history matching of water production rates in the facies estimation problem.

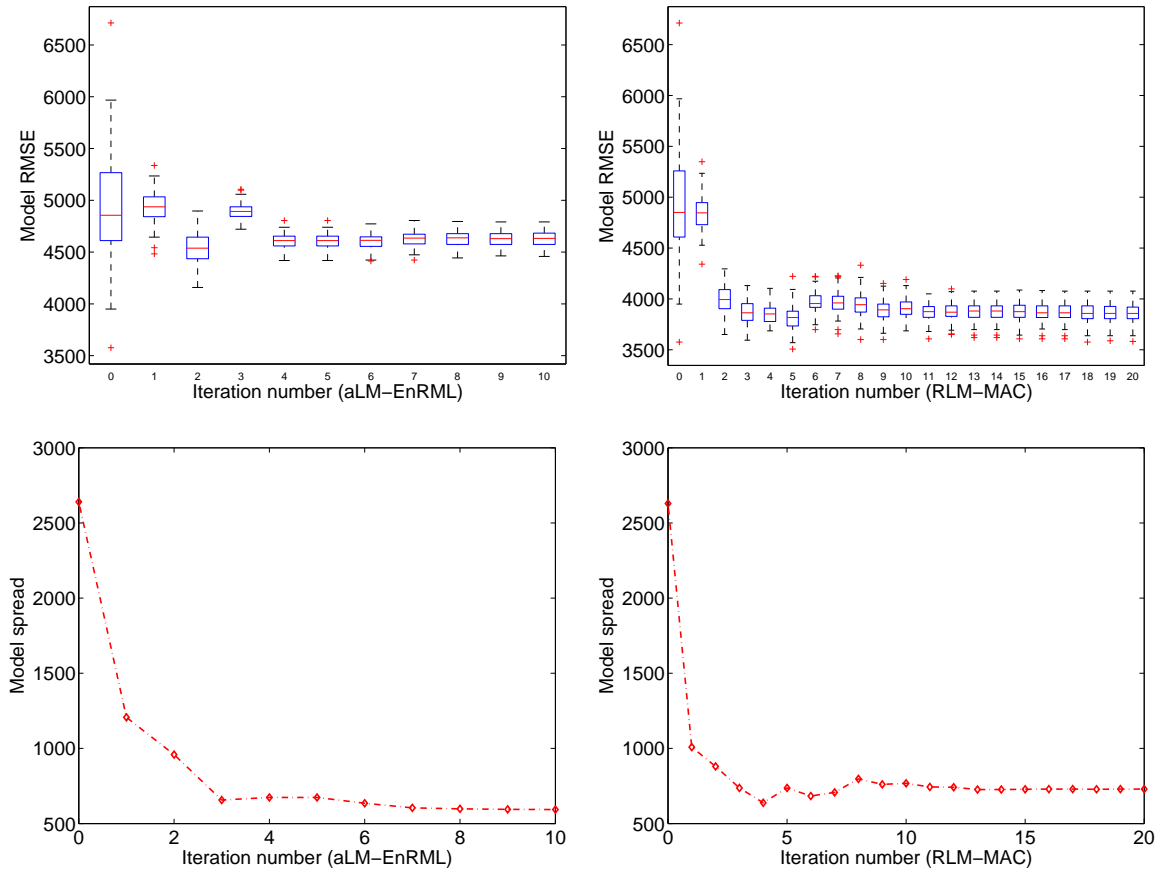


Fig. 13 Top panels: Box plots of RMSEs of the ensemble models—with respect to the reference model of the facies estimation problem—at different iteration steps in the aLM-EnRML (left) and RLM-MAC (right). Bottom panels: Ensemble spreads of the aLM-EnRML (left) and RLM-MAC (right) at different iteration steps.

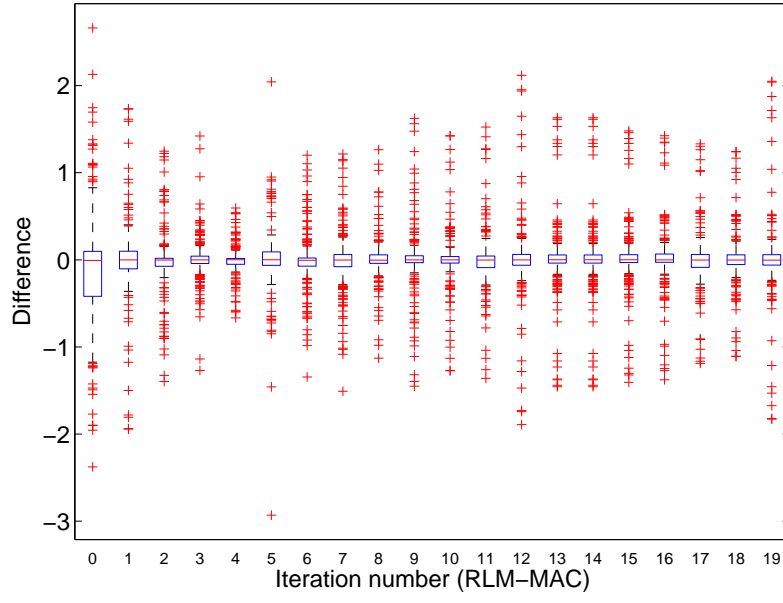


Fig. 14 Box plots of the normalized differences $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right)$ at different iteration steps of the facies estimation problem.

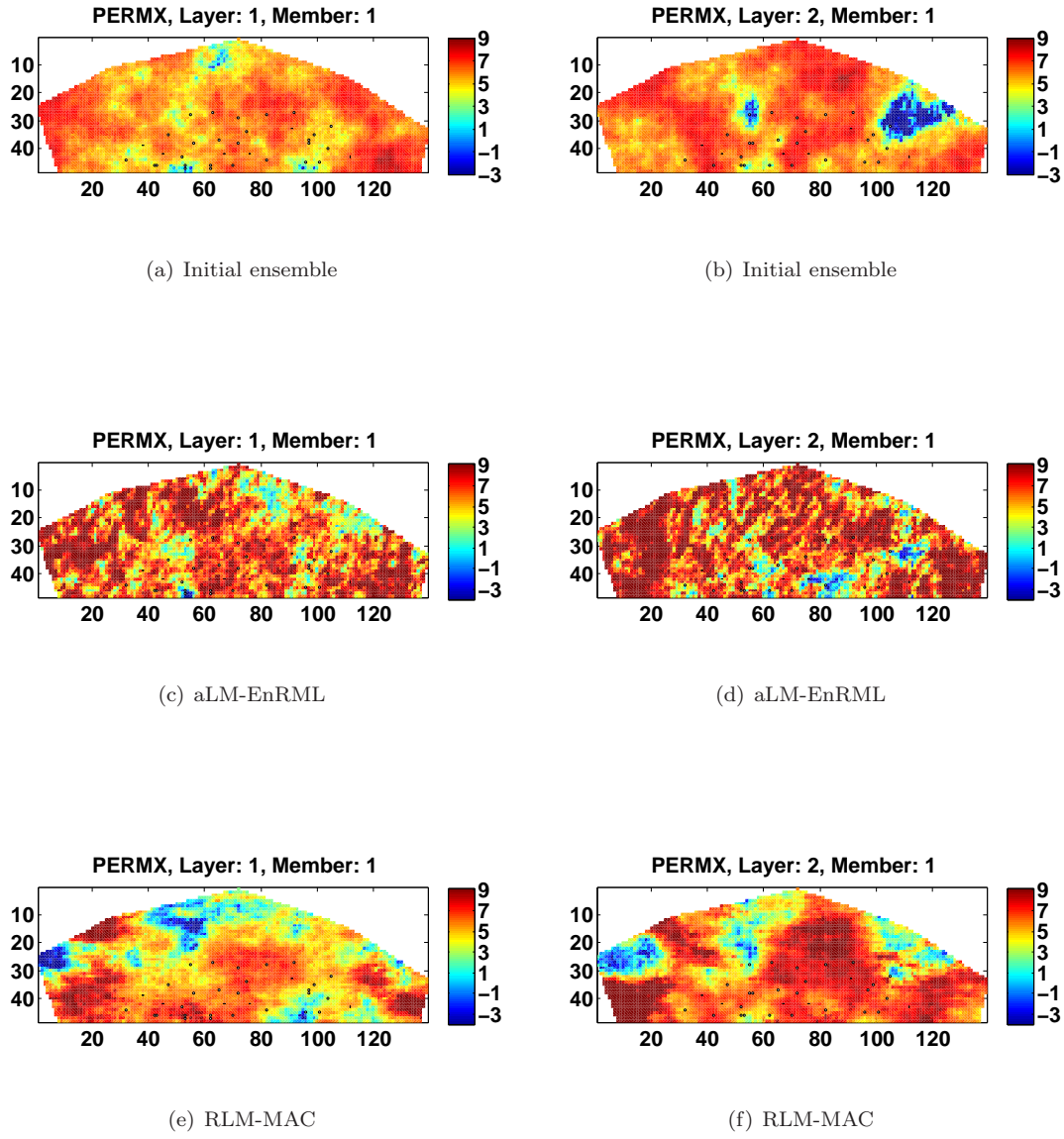


Fig. 15 Distributions of PERMX on layers 1 and 2 from a reservoir model of the initial ensemble (a,b), the corresponding reservoir model updated by the aLM-EnRML (c,d), and the corresponding reservoir model updated by the RLM-MAC (e,f). The dots in the figures indicate the locations of the producers and injectors in the Brugge field.

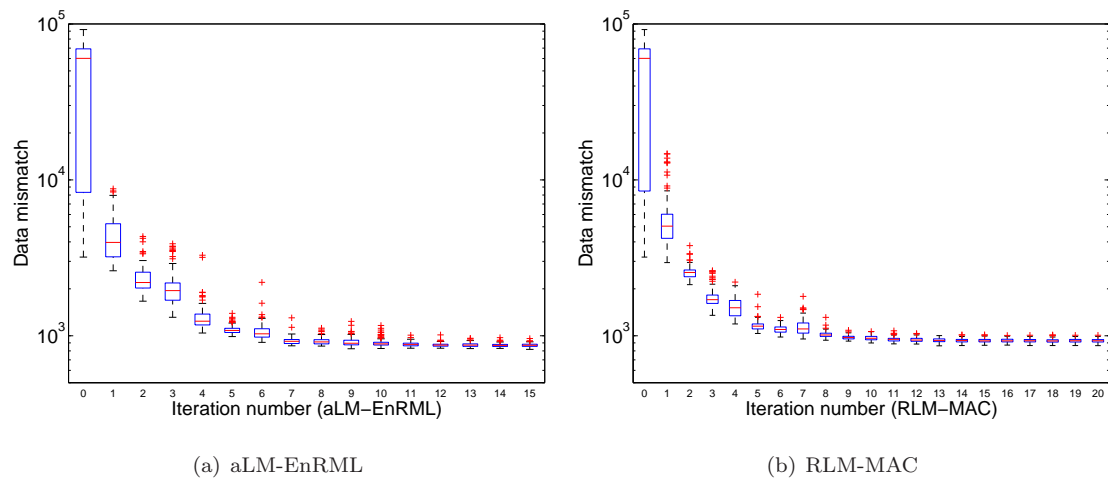


Fig. 16 Box plots of data mismatch of the aLM-EnRML (left) and the RLM-MAC (right) at different iteration steps in the Brugge field case, using production data at 20 out of 127 time instants in the first 10 years.

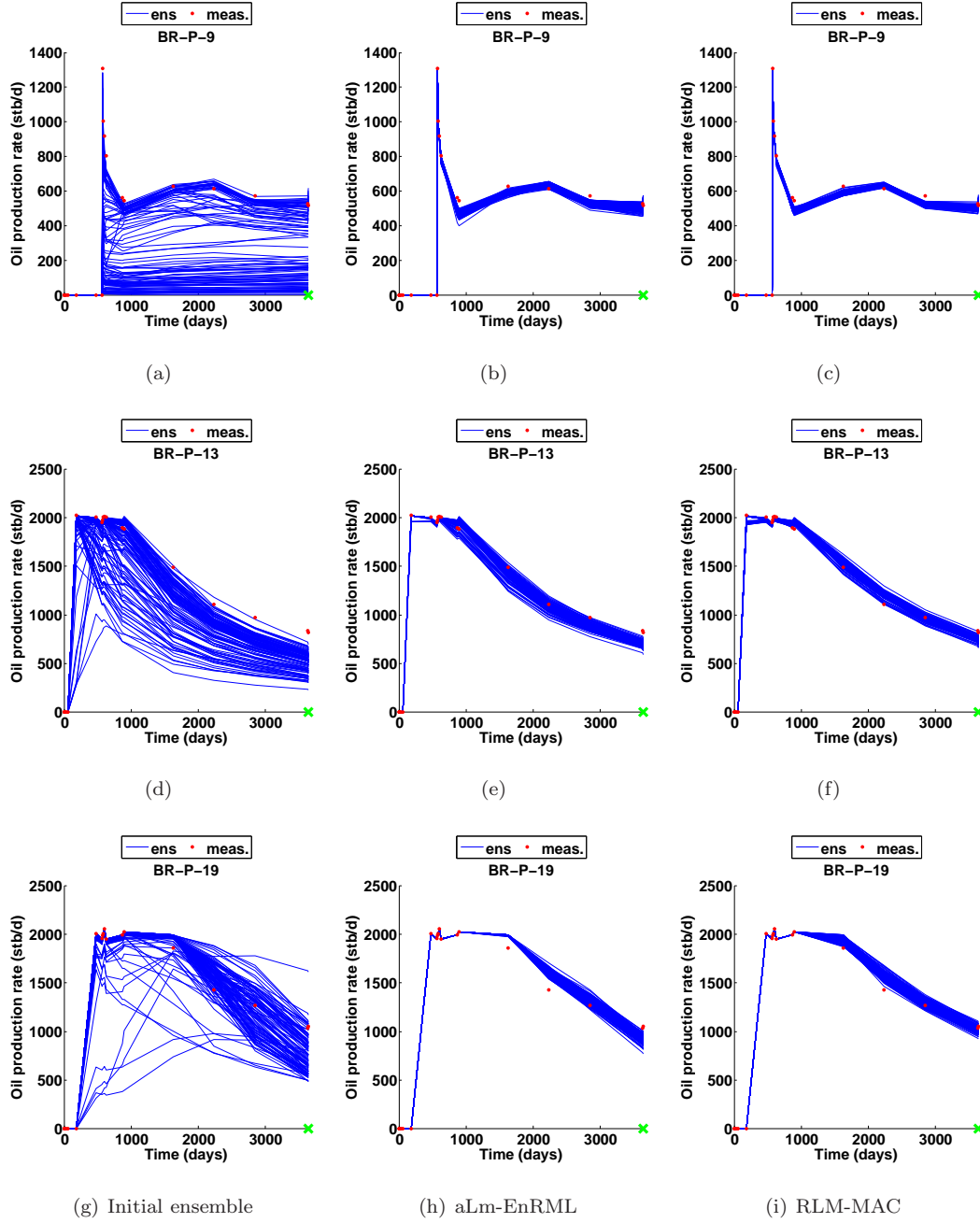


Fig. 17 History matching of oil production rates (WOPR) at BR-P-9 (top), BR-P-13 (middle) and BR-P-19 (bottom) in the Brugge field case, using the initial ensemble (1st column) and the ensembles of the aLM-EnRML (2nd column) and RLM-MAC (3rd column) at the final iteration steps. In each image, the red dots represent the historical WOPRH data and the blue curves are the forecasts with respect to the initial ensemble (1st column), and history matching profiles (2nd and 3rd columns).

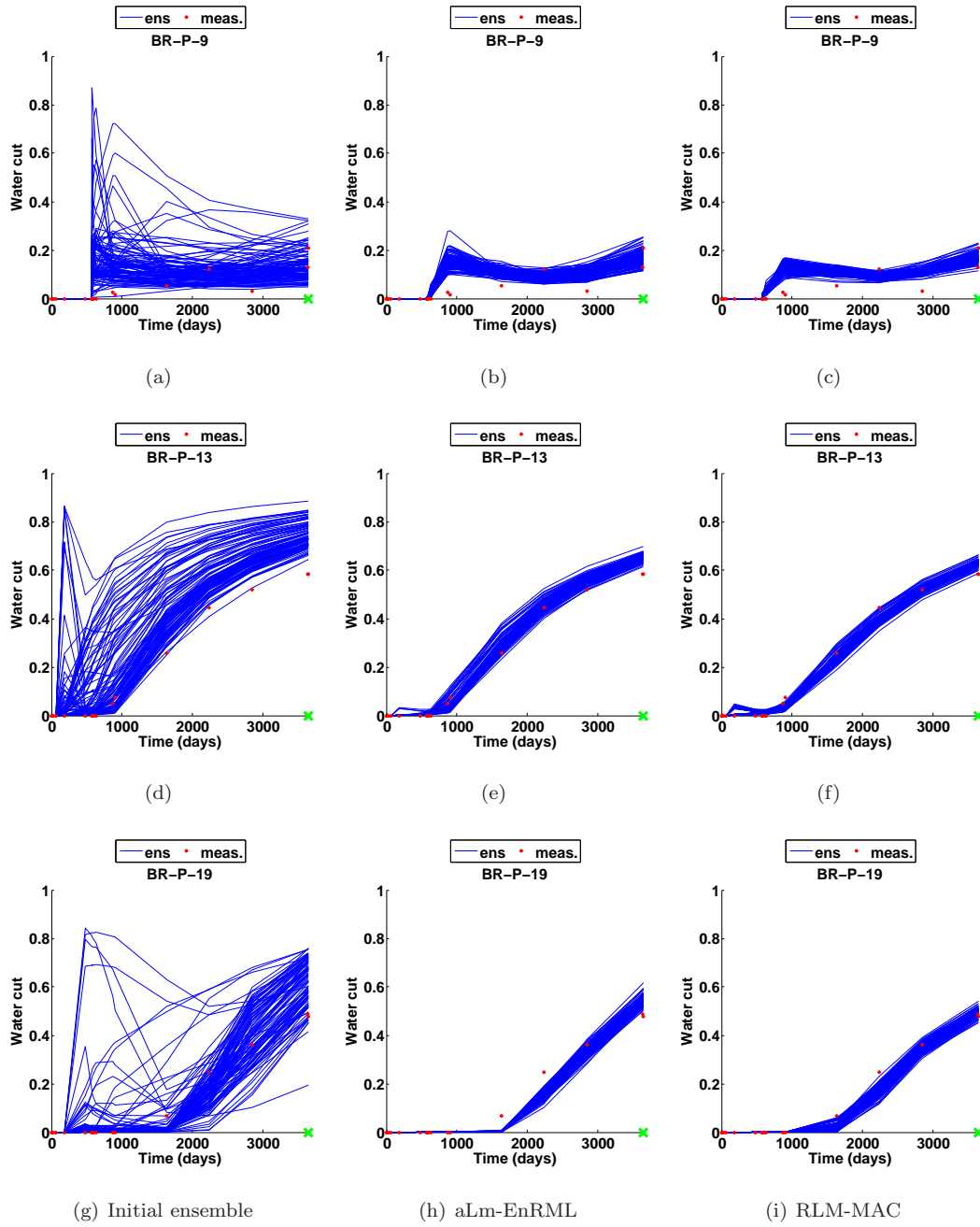


Fig. 18 As in Figure 17, but for water cut (WWCT) at BR-P-9, BR-P-13 and BR-P-19 in the Brugge field case.

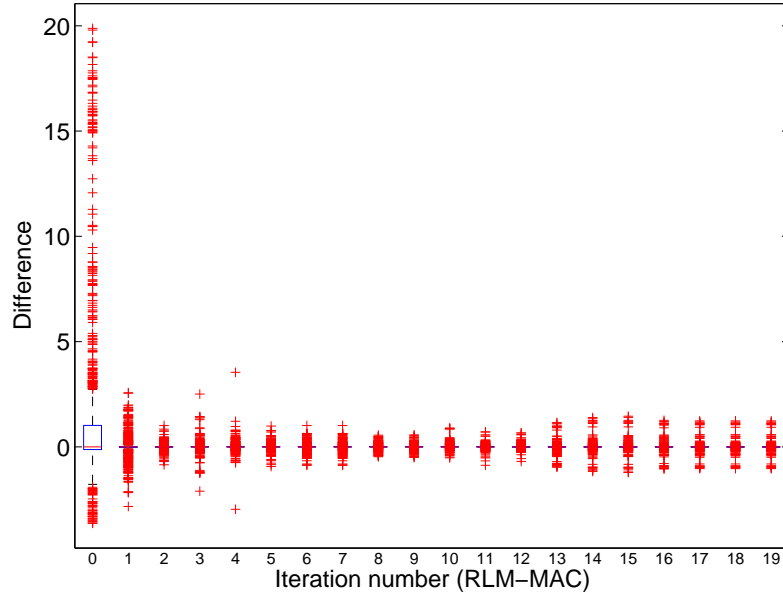


Fig. 19 Box plots of the normalized differences $\mathbf{C}_d^{-1/2} \left(\mathbf{g}(\bar{\mathbf{m}}^i) - \overline{\mathbf{g}(\mathbf{m}_j^i)} \right)$ at different iteration steps in the Brugge field case.

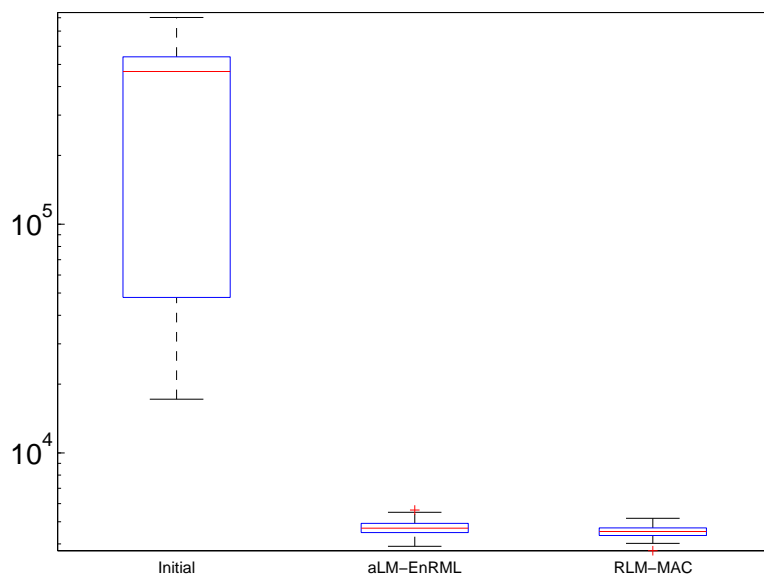


Fig. 20 Box plots of data mismatch with respect to the initial ensemble (left), the aLM-EnRML (middle) and the RLM-MAC (right) in the Brugge field case. Here, all the production data in the first 20 years are used to cross-validate the history-matched reservoir models obtained by aLM-EnRML and RLM-MAC using production data at 20 out of 127 time instants in the first 10 years.

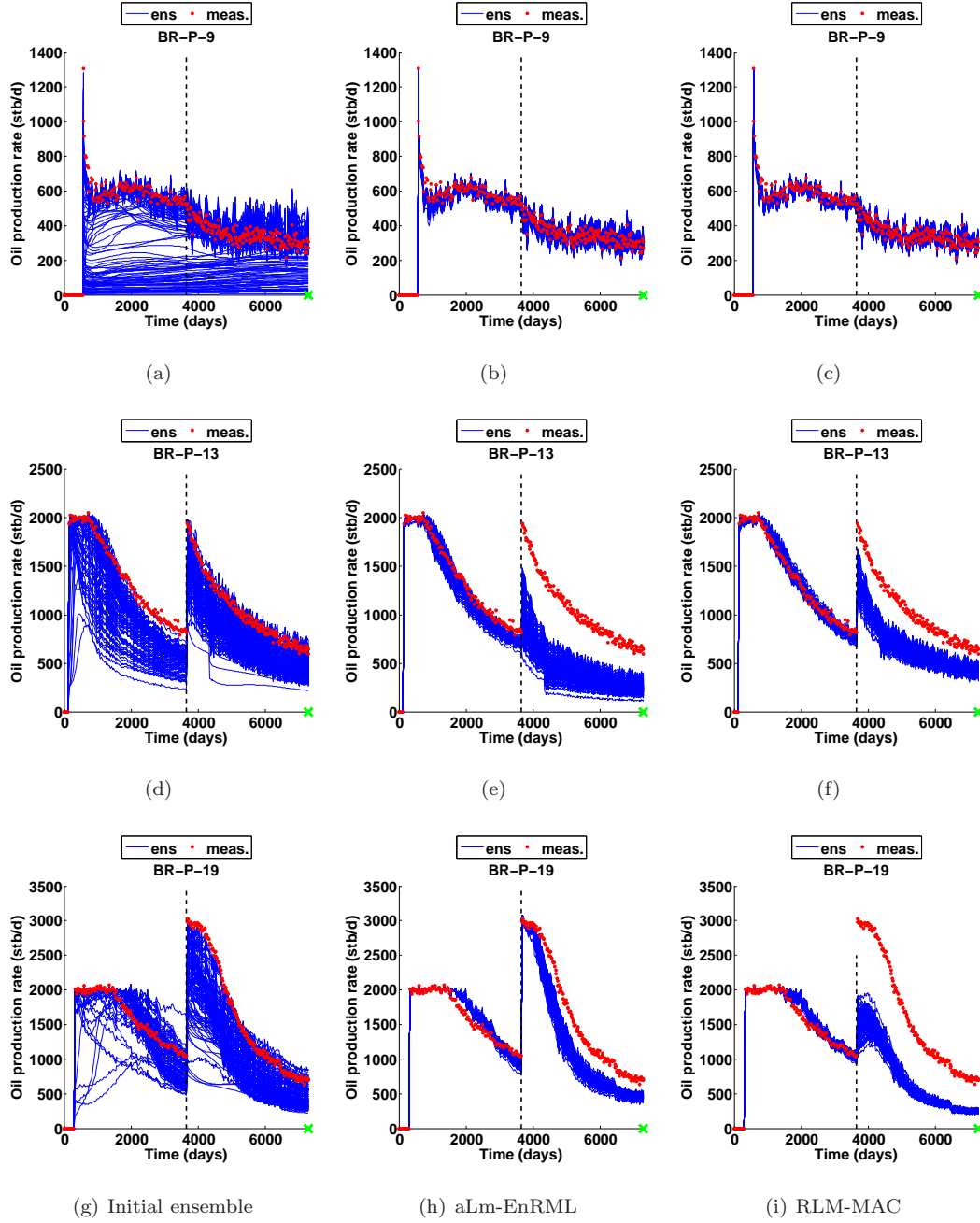


Fig. 21 Cross validation of oil production rates (WOPR) at BR-P-9 (top), BR-P-13 (middle) and BR-P-19 (bottom) in the Brugge field case, using the initial ensemble (1st column) and the ensembles of the aLm-EnRML (2nd column) and RLM-MAC (3rd column) at the final iteration steps. In each image, the red dots represent the historical WOPRH data and the blue curves are the forecasts with respect to the initial ensemble (1st column) and the final ensembles obtained by both iES (2nd and 3rd columns). The vertical dashed lines in the figures of the 2nd and 3rd columns separate the first and second decades.

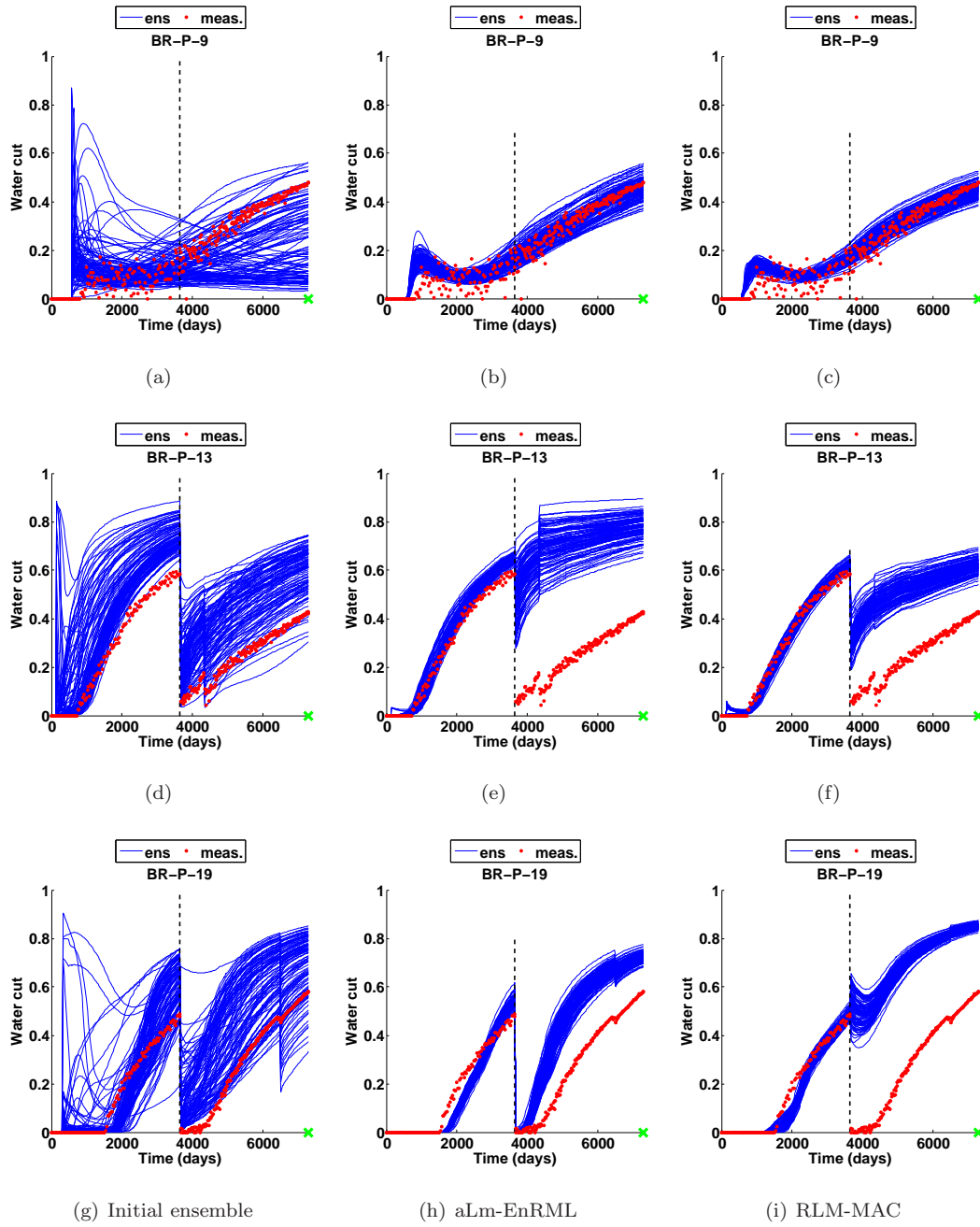


Fig. 22 As in Figure 17, but for water cut (WWCT) at BR-P-9, BR-P-13 and BR-P-19 in the Brugge field case.

This figure "L96_pie_rlm_obj.png" is available in "png" format from:

<http://arxiv.org/ps/1505.01135v2>

